

**COVENTRY UNIVERSITY**  
**Faculty of Engineering and Computing**

**PSOnline**  
**Primary School Online**

by

**Maciej Lipinski**

Student ID: 2009065

**393CS Computing Project**  
April 2008

Supervisor: **Richard Gatward**

**Declaration of Originality**

This project is all my own work and has not been copied in part or in whole from any other source except where duly acknowledged. As such, all use of previously published work (from books, journals, magazines, internet, etc) has been acknowledged within the main report to an item in the References or Bibliography lists.

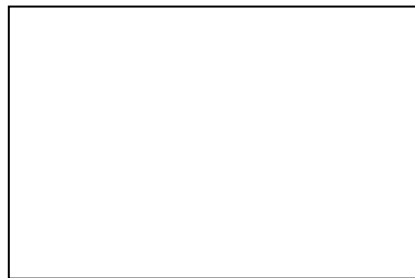
I also agree that an electronic copy of this project may be stored and used for the purposes of plagiarism prevention and detection.

**Copyright Acknowledgement**

I acknowledge that the copyright of this project and report belongs to Coventry University.

Signed:

Date:



Office Stamp

## Abstract

The aim of this document is to describe the process of development of POnline Project. The project's aim was to create a virtual online primary school administration system. A system to simplify communication and information exchange between the school (teachers and administration) and its 'clients' (pupils and pupils' parents) as well as within the school (teachers – administration).

The document describes the approach adapted to perform background research, requirement acquisition, design, implementation, feedback gathering and testing of the system and provides the information concerning the management of POnline project.

# Table of Contents

<b>Abstract</b> .....	<b>3</b>
<b>Table of Contents</b> .....	<b>4</b>
<b>Additional Materials on the Accompanying CD</b> .....	<b>7</b>
<b>Acknowledgements</b> .....	<b>8</b>
<b>1. Introduction</b> .....	<b>9</b>
<b>1.1 Background to the Project</b> .....	<b>9</b>
<b>1.2 Project Objectives</b> .....	<b>10</b>
<b>1.2.1 Gather requirements</b> .....	<b>10</b>
<b>1.2.2 Analyse similar systems</b> .....	<b>10</b>
<b>1.2.3 Design and implement prototypes</b> .....	<b>10</b>
<b>1.2.4 Gather intended users' feedback</b> .....	<b>10</b>
<b>1.2.5 Analyse intended user's feedback and redesign the system</b> .....	<b>10</b>
<b>1.2.6 Produce the product</b> .....	<b>10</b>
<b>1.2.7 Evaluate the product</b> .....	<b>10</b>
<b>2. Project Management</b> .....	<b>11</b>
<b>2.1 Project stakeholders</b> .....	<b>11</b>
<b>2.2 Requirements</b> .....	<b>12</b>
<b>2.3 Deliverables</b> .....	<b>12</b>
<b>2.4 Risk management</b> .....	<b>13</b>
<b>2.5 Project Schedule and Motivation</b> .....	<b>14</b>
<b>2.6 Scope</b> .....	<b>15</b>
<b>3. Background research</b> .....	<b>16</b>
<b>3.1 Literature review</b> .....	<b>16</b>
<b>3.2 Similar system analysis</b> .....	<b>17</b>
<b>3.3 Analysis of software used by the client</b> .....	<b>18</b>
<b>3.4 Analysis of Hardware</b> .....	<b>18</b>
<b>3.5 Technology review, comparison and choice justification</b> .....	<b>18</b>
<b>4. Methodology</b> .....	<b>20</b>
<b>4.1 Chosen methodology</b> .....	<b>20</b>
<b>4.2 Requirements gathering</b> .....	<b>22</b>

4.3	Requirements.....	23
4.3.1	Requirements by the users .....	23
4.3.2	Requirements imposed by clients' software.....	25
4.3.3	Hardware Requirements .....	25
4.4	Optional requirements.....	25
4.5	Requirements prioritisation .....	26
4.6	Feedback acquisition .....	26
4.7	Analysis.....	29
4.7.1	Use case diagrams.....	29
4.7.2	Sequence diagrams .....	31
4.7.3	Database.....	33
5.	Design .....	35
5.1	System architecture .....	35
5.2	Software architecture .....	36
5.3	Database design .....	37
5.4	Prototypes design .....	39
5.4.1	Prototype 1 design.....	41
5.4.2	Prototype 2 design.....	46
5.4.3	Prototype 3 design.....	48
5.5	Final design.....	50
6.	Implementation.....	51
6.1	Hardware used during the development.....	51
6.2	File organization .....	51
6.3	Database implementation .....	53
6.4	Prototypes implementation.....	54
6.4.1	Prototype one.....	54
6.4.1.1	Feedback .....	56
6.4.2	Prototype two .....	56
6.4.2.1	Feedback .....	59
6.4.3	Prototype three.....	60
6.4.3.1	Feedback .....	61
6.5	Final product.....	61
7.	Test.....	62
7.1	Test conduction.....	62
7.2	Test summary .....	64

<b>8. Conclusions.....</b>	<b>65</b>
<b>8.1 Achievements .....</b>	<b>65</b>
<b>8.2 Future Work .....</b>	<b>65</b>
<b>8.3 Problems encountered .....</b>	<b>65</b>
<b>8.4 Project Evaluation .....</b>	<b>66</b>
<b>List of References .....</b>	<b>67</b>
<b>Bibliography.....</b>	<b>68</b>
<b>Appendix A – Project management .....</b>	<b>69</b>
<b>A-1. WBS .....</b>	<b>69</b>
<b>A-2. Project Schedule .....</b>	<b>70</b>
<b>A-3. Project.....</b>	<b>71</b>
<b>Appendix B – Liaison with the client and users .....</b>	<b>72</b>
<b>B-1. Report form initial interview with the client .....</b>	<b>72</b>
<b>B-2. Report from the second interview with the client .....</b>	<b>73</b>
<b>B-3. Collective report form interviews with a rep. of teachers, parents and pupils ..</b>	<b>74</b>
<b>B-4 Example how the timetable changes are displayed by the client currently .....</b>	<b>75</b>
<b>B-5. Table with prioritized requirements .....</b>	<b>76</b>
<b>B-6 Screen shot of the online questionnaire with description .....</b>	<b>77</b>
<b>Appendix C – Instruction on POnline installation.....</b>	<b>78</b>
<b>C-1: Login screen .....</b>	<b>78</b>
<b>C-2: Web Diagram of teacher interface .....</b>	<b>79</b>
<b>C-3: Example screenshot of administrator interface with descriptions.....</b>	<b>80</b>
<b>C-4: Information about PHP-ExcelReader .....</b>	<b>81</b>
<b>Appendix D – POnline installation instruction .....</b>	<b>82</b>
<b>Appendix E – test .....</b>	<b>86</b>
<b>Appendix F – Critical evaluation by the client .....</b>	<b>89</b>

***Additional Materials on the Accompanying CD*****DVD Contains:**

- **Final Report:** Lipinski\_M\_2008.pdf

**and the following folders:**

- **Data to import** – contains:
  - sample import data to test the import mechanism
- **Design**– contains:
  - materials used during system design
- **Implementation**– contains:
  - materials used during the implementation
- **online survey results**– contains:
  - prototype two, online survey and the survey results
- **PSONline Pack**– contains:
  - the Pack including: PSONline (final version), instruction how to install it, WAMP5, sample data – this pack was sent to the user for acceptance tests.
- **Requirements**– contains:
  - Reports from consultations with the client
  - Sequence Diagrams
  - Use Case Diagrams and Descriptions
- **Research**– contains:
  - analysis of online primary school systems
- **Screen shots**– contains:
  - sample screen shots of the of PSONline interface
- **Tests**– contains:
  - approval tests
- **PSONline - Final Product** – contains:
  - The final version of the PSONline System

## **Acknowledgements**

I am very grateful to the client for the extremely good cooperation, the work and time he devoted to the project.

Many thanks to my housemate for consultations and unlimited patience.



## **1. Introduction**

### **1.1 *Background to the Project***

The client is Primary School number 22 in Rybnik, south of Poland. The school is innovative and tries to be up to date with new technologies. It cooperates with many foreign primary schools i.e. in Austria, Norway, Slovakia, and the school's teachers attend UE-founded international programs visiting primary schools around the EU. The ideas and solutions observed abroad, which are found useful and possible to adapt to Polish reality, are implemented in the school. The system which allows parents, pupils and teachers to see most important information over the Internet was spotted by the headmaster of the school while visiting schools in Portugal (Coimbra) and United Kingdom (London). The idea was recognized as a good way to sustain an active and frequent liaison with parents. Although a decrease in involvement of parents in their children school life has been noticed, there was no sufficient funds to buy such a system. Thus, my offer to create the system within my Bachelor degree project was met with enthusiasm and instant approval.

## **1.2 Project Objectives**

### **1.2.1 Gather requirements**

Investigate the client's and intended users' vision of the project and establish their precise requirements concerning the project.

### **1.2.2 Analyse similar systems**

In order not to "re-invent the wheel", analyse similar systems to establish well defined solutions to most common problems, find out the most troublesome areas of project and avoid mistakes made by the others.

### **1.2.3 Design and implement prototypes**

Design and implement prototypes of limited functionality to establish client's and intended users' 'real' requirements. Build each subsequent prototype on the previous prototype changed according to users feedback.

### **1.2.4 Gather intended users' feedback**

Use appropriate means (interviews, surveys, consultations) to collect intended users' feedback in order to specify their needs and their requirements.

### **1.2.5 Analyse intended user's feedback and redesign the system**

Analyse users' feedback to establish clear requirements. Change the system to reflect users' reasonable requirements.

### **1.2.6 Produce the product**

After satisfactory number of prototyping iterations obtain a final product that satisfies client's needs.

### **1.2.7 Evaluate the product**

Critically evaluate the project and indicate the areas of further improvements.

## 2. Project Management

*Project Management is defined as "...the application of knowledge, skills, tools and techniques to project activities to meet project requirements."*

*(Project Management Institute, 2000)*

Good project management has to incorporate the following elements:

- Recognition of project stakeholders,
- Clear requirements,
- Thorough risk analysis,
- Clear deliverables,
- Good planning and time management,
- Motivation.

This chapter evaluates management of the project according to above mentioned elements.

### 2.1 Project stakeholders

*"These are people who have a stake or interest in the project. It is important that they be identified as early as possible"* Cotterell, M. & Hughes, B. (1995: 11)

Table 2.1-1 shows stake holders that were identified and their involvement assessed according to the chart presented in Figure 2.1-1

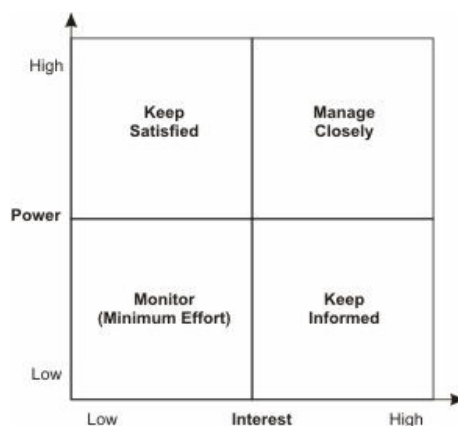


Figure 2.1-1

Stakeholder or stakeholder group	Involvement	Power and interest
Client	The system is his idea. He wants the school to be more “user-friendly” and modern. He intends to improve the quality of school by introducing the system. He provides the most of the requirements and feedback.	HH
Pupils	The system is suppose to make school-related information easily available for them	ML
Parents	They provide feedback to the designed system	LL
Teachers	The system will add to their duties the necessity of entering grades. The system will be an easy tool for them to communicate with parents and will help in with their work. They provide feedback to the designed system	MM
Supervisor	His duty is to asses the outcome of the project and the performance through the project as well as give assistance.	HH
Second assessor	Assess(along with the supervisor) assessed elements of the project	HL

Table 2.1-1

## 2.2 Requirements

Requirements can be divided into two groups:

- 393 CS Computing Project module defined by the University and investigated at the commence of the project:
  - Delivery of the Report Brief within deadline,
  - Constant liaison with the supervisor,
  - Oral Presentation,
  - No plagiarism,
  - Delivery of the final report and project with the deadline,
- Requirements concerning the final product of the project specified by the client and intended users – acquired throughout the project and discussed in details in Chapter **4.3 Requirements**.

## 2.3 Deliverables

- Product, constitute of:
  - System engine written in PHP providing each pupil, teacher, parent with individual account, delivering appropriate information to targeted recipients, providing compatibility with other software used by the client.
  - SQL database compatible with other databases used by the client
  - Graphic interface diversified regarding the user’s group
  - Technical documentation and user manual
- Report, consisting of:
  - Background research into the intended users group needs and similar projects

- An investigation into:
  - hardware to be used
  - software used by the client to assure system's compatibility
- Project management documentation
- A critical evaluation of the product and the project

## 2.4 Risk management

The following risks connected with the project were recognized and assessed according to their impact on the project and their probability.

Risk	Probability (L-M-H)	Impact (L-M-H)
<b>Risk on the initial state:</b>		
Wrong choice of the project	L	H
No client found	M	M
Wrong assessment of the skills needed for particular project	M	H
<b>Risk on the research state:</b>		
No or poor background research	M	M
Wrong technology choice	M	H
Wrong tools choice	M	M
Wrong methodology choice	L	L
Poor requirement definition	M	H
<b>Risk on design state:</b>		
Choosing a wrong approach to the problem	M	H
Spending too little time on the design, disregarding the significance of good design	M	M
Wrong design	L	H
<b>Implementation state:</b>		
Poor or no experience in the tools or methodologies used	H	M
Cheating	L	H
Finding mistakes in the design	H	L
Finding problems impossible to solve	L	H
<b>Test state:</b>		
Discovering logical bugs in the project design	L	H
Coming across unfixable bugs	L	H
<b>Report writing:</b>		
Leaving the report writing to the last minute	H	H
Plagiarism	L	H
<b>General risk:</b>		
Data loss	L	H
Laziness	H	M
Bad luck	L	H

Table 2.4-1

To minimize the risk the following means were taken:

- Careful planning,
- Time buffers – periods of time at the end of tasks and before milestones left unscheduled in the case of unpredicted situations,
- Clearly set milestones and the fact that meetings (online meetings) with the client were arranged in advance and thus enforced keeping with the schedule.

## 2.5 Project Schedule and Motivation

After investigating initial requirements and conducting of the project a basic work breakdown structure (WBS) was produced.

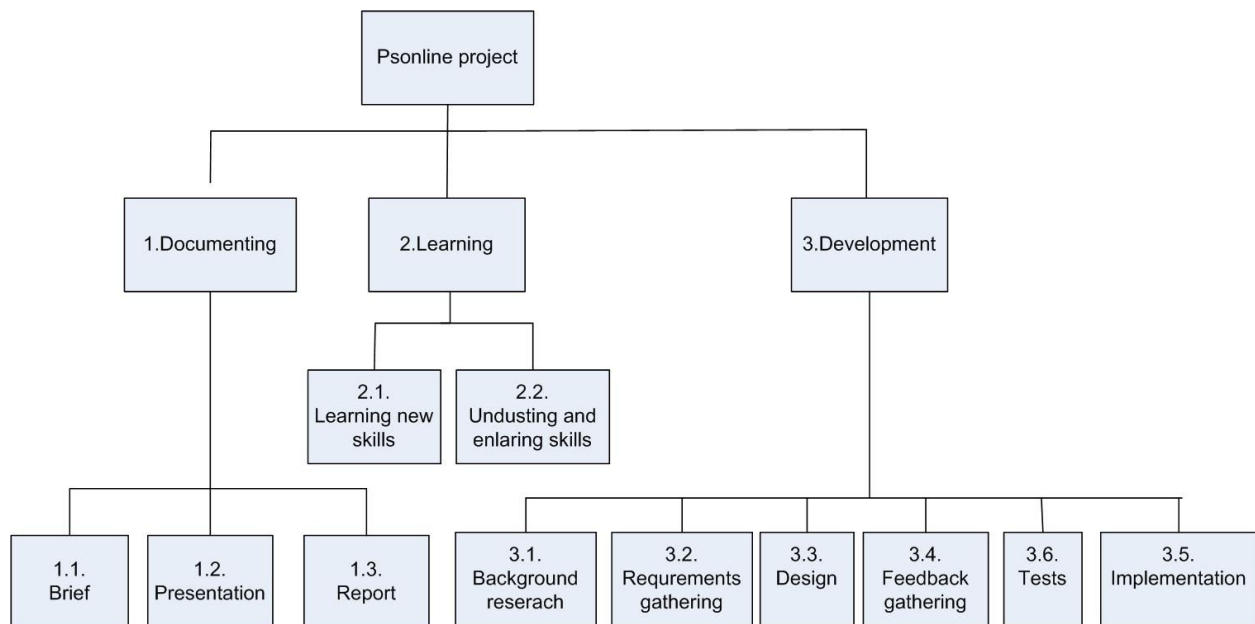


Figure 2.5-1

The WBS was used to create a schedule which was presented to the project supervisor. The schedule introduced the following general milestones:

- Brief submission,
- Oral presentation,
- Final report submission,
- Finish of background research – start of design,
- Finish of design – start of implementation,
- Finish of implementation – start of evaluation,

However, it had to be rearrange significantly when the methodology was chosen and the decision to conduct an online survey made. Thus a new, more detailed WBS and new schedule were produced (**Appendix A: A-1,A-2**)

To keep a constant and high level of motivation throughout the project, the project's plan, in a modified form, was presented to the client (**Appendix A: A-3**). The most important were the milestones determining finishing dates of each prototype. In advance appointments were made with the client to acquire feedback on each

prototype. In the consequence the schedule had to be kept. Any delays would be very troublesome and could undermine good liaison with the client.

## 2.6 Scope

The scope of the project is to deliver fully functional system which enables exchange of school related information (grades, timetable and its changes, personal data) between pupils, parents, teachers and administration. The system should be easily accessible by users from their homes

Access to information should be strictly controlled and limited only to the information needed for the users. Therefore, each user should have her/his own account with the interface and information relevant to the group affiliation (pupils, parents, teachers, administration). Ex. pupils should have access to the appropriate timetable and its current changes if any; names of pupil's teachers, classmates and grades but not to teachers' timetables and teachers' data.

The system should be compatible with client's software enabling import of appropriate data.

The development of the system should be appropriately documented in the Final Report document

The project should be finished within the time constraints.

The scope of the project does not include

- Encryption protocol, ex. Security Socket layer,
- Internal message exchange between users,
- Exchange and storage of any other information than: grades, timetable, timetable changes, users' data, information about classes.

### 3. Background research

#### 3.1 Literature review

The idea of supporting education with Information and Communication Technologies (ICT) has been affecting the education process severely for many years. Although the main effect of introducing ITC is on the higher education, the notion of using ITC in primary education was a topic of the study held on the request of European Parliament as long as 10 years ago. The study by **PJB Associates(1997)** recognized that *“there is a need to enable software producers to develop an awareness of the range of learning styles deployed in primary and secondary schools.”* Since then a great development has been made in ITC for primary education. The reason for that was mentioned few years later in the article by **Körös-Mikis, M.(2002)**. She states that innovative pedagogical practice based on ICT helps in the transformation of society to become information-based, and can be especially helpful in industrial societies transformation. Nowadays ITC based education is something natural and desirable in primary education. Although the developed system is not an e-learning application, it is part of the general trend towards wide ITC development in education and can definitely be a good motivation for pupils to develop ITC skills.

Although the idea of putting online the information about the progress of primary school pupils seems quite innovative in Poland (at least), the general idea is well established at universities. It is worth noticing that the difference between university online information systems and the system developed in this project is only about the users and range of functionality but the idea is exactly the same.

Such systems are web applications, defined in **Booch, G. (2001)** article as *“Web site where user input, including navigation through the site and data entry, affects the state of the business”* (students enrol on courses, have online tests etc.). The article nicely presents the idea of web applications and its architecture. It points out that : *“A system's architecture represents the necessary strategic design decisions sufficient to form that system”*. However, the author clearly favours web applications technologies such as JSP or J2EE. The article comparing different web application technologies (**Corp21 2005**) shows that for the need of the developed project open source and less advanced technology can be used.

A thorough analysis of different client-server architectures was found in **Sommerville, I. (2007:397)**. The three-tier architecture is described there as used for *“Large-scale applications with hundreds or thousands of clients”,* where *“data and application are volatile”* and *“data from multiple sources integrated”*. The book can be really helpful in establishing good ground for architecture choice.

*“PHP has many advantages over similar technologies, namely high efficiency, interfaces to many databases, many solutions for common WWW tasks, low cost, simplicity in learning and using, offers good object orientated mechanism, is platform independent, there are many free source codes and support is available”*. This statement by **Welling, L. & Thomson, L.(2005:27)** was being evaluated through the project development.

The knowledge about databases, their creation, optimization, possible models was based on **Silberschatz, A. (2001)**.



### 3.2 *Similar system analysis*

In order not to “reinvent the wheel”, not to repeat someone else mistakes and to pick up good ideas it was decided to analyse similar systems.

The research was limited mainly to websites of Polish schools and Polish systems, as the client is a school in Poland, thus the intended users are Polish and the educational system is specific for Poland.

The following system were analysed in details:

- System implemented in Primary School number 3 in Zielonka
- Sofokles – commercial system which was offered to the client
- Dziennik ucznia – commercial system which was offered to the client

Detailed analysis are included on the **CD (Analysis of online primary school systems)**, below are presented the most important findings of the research:

- Solutions which seemed good for implementation:
  - User having to change automatically generated password when entering the system for the first time,
  - Calculating pupils' grade average and class grade average,
  - Usage of predefined categories of grades and possibility to define new categories,
  - The ability to upload by the user his/her photo,
  - Possibility of assigning descriptive grades, which is a requirement in levels 1-3.
  -
- Solutions worth considering implementation:
  - The possibility to write a mark to all the pupils in one class,
  - Storing data concerning pupils' absence/presence,
  - PESEL number (Universal Electronic System for Registration of the Population) to be user's login for teachers and parents,
  - Access to data from previous years.

### 3.3 *Analysis of software used by the client*

The client uses software provided by the Polish company called VULCAN, the biggest school-management application provider of in Poland. None of the software application used by the client provides all of the functionality desired by my client.

The data which has to be imported from abovementioned packages:

- Teachers, pupils and parents data,
- Timetable,

The interest of the research was:

- available formats of exported data,
- structure of data and information included,

The research outcomes:

- Timetable – “**Plan lekcji Optimum**” offers the possibility to create standalone website containing timetable,
- Pupils and parents data can be easily exported to XML by the “**Sekretaria Optimum**”,
- Teachers’ data – export from “Arkusz Optimum” to Exel file

### 3.4 *Analysis of Hardware*

The school is connected to the Internet with 1MB ADSL line (1:4 – upload : download). This kind of connection is not good for handling multiple simultaneous requests to the sever. If a server is to be created at school, the connection has to be changed to DSL or most preferably separate connection devoted only to the server has to be installed.

However, a better solution would be to buy a server space on a professional server.

The advantages of such a solution:

- Allow access by multiple users simultaneously (depending on the agreement),
- The security and fulfilling law regulations concerning data protection is guaranteed by the provider,
- The maintenance is guaranteed.

### 3.5 *Technology review, comparison and choice justification*

The task of creating a “system” to allow online data exchange between pupils, parents, teachers and school administration can be approached in two significantly different ways:

- Ordinary application
- Web application

Obviously the Web application is more appropriate choice for the intended product. It is platform independent, allows easy improvements or system changes and possibility to use the system from any computer connected to the Internet.

Web applications are commonly structured in three tiers:

1. first tier – web browser,
2. middle tier – engine using dynamic web content technology,
3. third tier – database,

The choice has to be made of the technology to be used for system engine and database which are closely related and affects each other.

Below table presents cooperation of five most common technologies for web application development.

Technology Name	Sponsor	Operating System	Server	Data base	Lang	Costs	Example Pages
CGI	Open source	Linux Unix	Apache	Varies	Perl	Free	
ColdFusion	Marcomedia	Windows Linux Unix	ColdFusion	Varies	CFML	expensive compared to some of its competitors	
LAMP	Opens Source	Linux Windows Unix	Apache	MySQL	PHP	Free	Yahoo.com Facebook.com
Java/J2EE	SUN, IBM	Linux Unix	J2EE	Varies	Java	Some are free	
.NET	Microsoft	Windows	ASP.NET	SQL server	VBasic C#	Technology free but MS Visual Studio, IIS and MS Access	Coventry.ac.uk

Table 3.5-1 Web applications technology cooperation (**Corp21 2005**)

The choice of the Web application technology is usually made according to the following criteria:

- Size of web page,
- Available budget,
- Operating system preference,

As the client is a state institution in Poland, with budget is too limited to support any cost-demanding technology, the open source free technology has to be chosen, with free tools for development and maintenance. However, all schools in Poland use MS Windows as the operating system and running Linux or UNIX can be troublesome and discouraging for the client. The best solution is an open source free server and free database which can be installed on the MS Windows. The easiness of installation, reinstallation and maintenance is crucial as it reduces the costs. This is why WAMP approach has been chosen: Windows, Apache, MySQL, PHP. Moreover the WAMP5 has been recommended. It is a Windows web development environment which allows to create web applications with Apache, PHP and the MySQL database.

## 4. Methodology

### 4.1 Chosen methodology

The methodology chosen to develop PSONline is software prototyping, however it does not fit precisely into any of the prototyping types.

To choose the methodology, the following project features had to be taken into account :

- Age of the pupil user group – 7 to 13. It would be difficult to explain in word to the youngest users, when acquiring feedback, what is the system going to do, Therefore, any feedback would be impossible. Once they can play with the system prototype, it is possible that a useful feedback can be acquired,
- It is much more attractive for users to give feedback on working application,
- The client could see how the work progresses despite being in Poland.

Implementation of the methodology involved creating prototypes which displayed necessary characteristics to obtain users' requirements -rapid prototyping (Computerworld 2002). However the prototypes were not discarded, they were changed to meet users requirements and used to create subsequent prototypes - evolutionary prototyping (**Sommerville, I. 2007**).

The general idea of the methodology is depicted in the Figure 4.1-1 – each iteration (called Phase) constitute of four main steps.

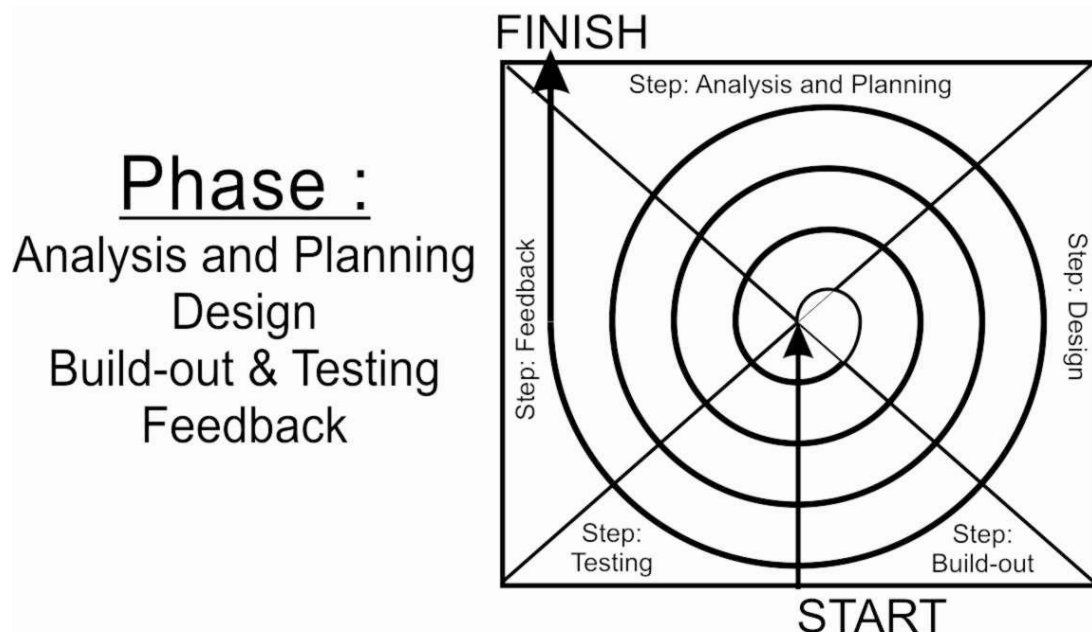


Figure 4.1-1 Prototyping methodology

The methodology implementation is presented in the Figure 4.1-2 below.

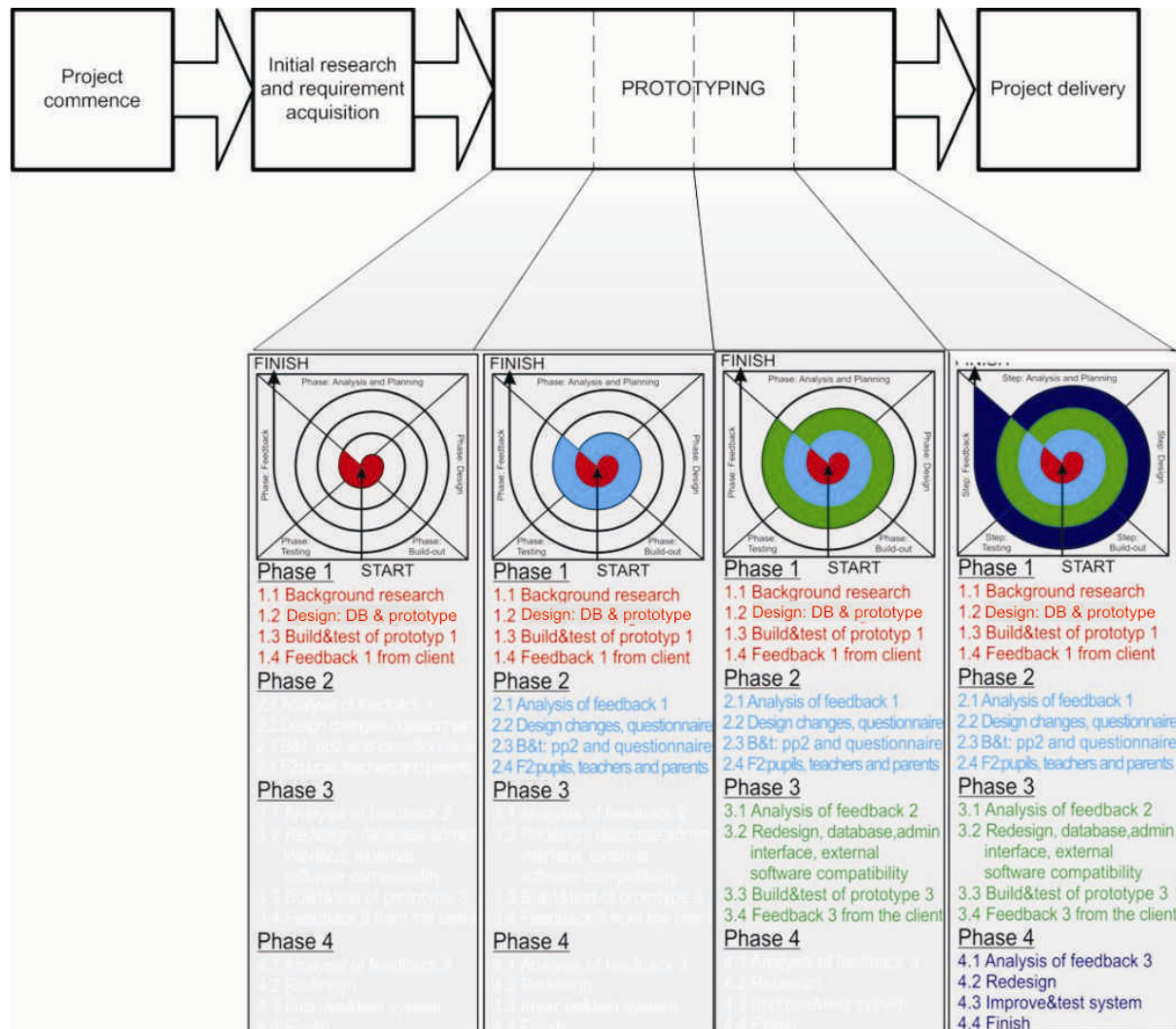


Figure 4.1-2 Implementation of prototyping methodology

Due to the methodology choice, especially during prototyping the requirement gathering can be easily confused with feedback acquisition. Thus the following distinctions between these two processes applies:

*Requirements are gathered at the beginning of each prototyping phase and cover the part of the system which is to be designed.*

*Feedback is acquired after the completion of each prototyping phase (or sometimes during implementation). The feedback concerns the part of the project which has been implemented so far. The analysis of the feedback can bring new requirements to the further development.*

## 4.2 Requirements gathering

The requirements gathering is one of the most important parts of the project. Should the requirement be wrongly determined, the client receives the product which does not satisfy client's needs. Or, if the problem is discovered earlier, the project needs to be redesigned and changed. It is time and effort consuming, thus determining what the client needs is crucial.

The choice of gathering requirements techniques to be used in this project was made taking into consideration the following factors:

- Project nature – web application,
- Intended users – extremely varied group, ranging from children (7-13 years old) to part time retired teachers (60 year old),
- Localisation of the client (Poland) and a very limited opportunity for personal meeting with the client and representatives of intended user groups.

The following techniques are used through the project:

- Interview – is a main source of basic information about requirements. All the interviews had informal character. One interview were conducted with the mean of Skype. After each interview a report was prepared to collect the most essential outcomes of the interview (enclosed in Appendix B). The interviews were performed following the below tips from (<http://www.guydavis.ca/seng/seng611/techniques/>):
- Prototyping – this kind of data requirement acquisition is a consequence of the methodology chosen for the project development
- Consultation – the constant liaison with a client, which is essential, had a character of consultation. The forms of consultation: email, skype conversation or messenger conversation
- analysis

Upon the commence of the project an interview (**Appendix B: B-1**) with the client was held to specify what was the client's vision of the system and to receive very basic requirements. This information was needed for the background research, methodology and technology choice. During background research import data requirements were established by analysing clients software and its export capabilities. During the Christmas Break a series of interviews was conducted with representatives of all the intended user groups to gather detailed requirements for the first design. The interviews were held (in the following order) with the client and would be administrator, a representative of teachers, parents and a pupils (**Appendix B: B-2,B-3**).

Throughout each phase during the research step (sometimes design) consultations with client were conducted to specify the requirements concerning project development in the particular phase. (**Draft reports from the consultations can be found on CD**).

During each research phase (except phase 1) the feedback from the previous phase was analysed to find out whether requirements redefinition is needed or new requirements occurred.

## 4.3 Requirements

### 4.3.1 Requirements by the users

The requirements are presented in the SHALL Statement form. They are listed chronologically divided by each development phase to show how new requirements were added and how they changed through the development.

#### Initial requirements – Initial Interview (Appendix B: B-1)

The system **shall** have 4 user groups: administrator, pupils, teachers, parents. Each user group shall have different user interface, authority and system options available.

The system **shall** allow users to see school related information concerning the particular user.

The system **shall** be easy to use and easily available and accessed from users homes.

Any information in the system **shall** be accessed only after logging in.

The system **shall** allow easy import of required data from the software used by the client.

#### First prototyping phase – Interviews (Appendix B: B-2,B-3)

Pupils, teachers and parents **shall** have limited access to the information in the system, namely

- Pupils **shall** have access to their grades, the names of the teachers and pupils in the class the pupil attend, timetables of all the classes, timetable changes
- Parents **shall** have access to all the information that their children have access to
- Teachers **shall** have access to grades, the names of the teachers and pupils in the classes they teach, pupils' and parents' details in the classes they teach, timetables of all classes and teachers, timetable changes,
- Administrator **shall** have unlimited access to all the information.

The following data concerning grade **shall** be stored and processed by the system: grade, grade category, grade date, grade description, teacher who graded, pupil who was graded

#### First prototyping phase – consultation

Pupils and parents **shall** not have possibility to enter any data to the database

Each user **shall** see his/her login, name and surname when logged to the system

The following data concerning administrator **shall** be stored and processed by the system: User login, user password, surname, name, second name.

The database **shall** have all the features described in the Consultation 1 notes (**CD: consultation-phase 1**)

The grades **shall** be displayed in the way described in Consultation 1 notes (**CD: consultation-phase 1**)

#### First prototyping phase – feedback analysis

The system **shall** be in Polish (bilingual optionally).

### Second prototyping phase – consultation (CD: Consultation-phase 2)

Teachers **shall** have the possibility to access information about the grades the pupils has received in the subject the teacher teaches.

The language **shall** be chosen upon logging into the system.

The system **shall** impose on the user the grades grouping.

Initial grade categories **shall** be defined.

The system **shall** allow adding new grade categories.

Information about using the system, so called “help”, **shall** be possible to enter without logging to the system.

The system **shall** calculate pupils’ grade average and class grade average.

The system **shall** show current timetable changes in the way most similar to the way it is done currently at school (**Appendix B: B-4**).

The system **shall** allow the teacher to determine the class of which data is displayed.

The system **shall** allow the parent who have more than one children at school to determine which child data the parent wants to browse.

An account **shall** be created for each parent registered in school database.

### Second prototyping phase – feedback analysis

All the teachers **shall** have the access to all the other teachers detailed information, most importantly the contact details.

The system **shall** allow in levels 1-3 to enter the descriptive grades.

The system **shall** allow teachers to enter grades to more than one pupil at the same time – multiple add grade.

### Third prototyping phase – consultation (CD: Consultation-phase 3)

The system **shall** allow to write one kind of grade to all the pupils of a given class at once, with the same description, date and category affiliation (can be useful when grades for tests are inputted).

The administrator **shall** be able to add, edit and delete the following data

- all users’ detailed information (import or entering through system)
- timetable (import only)
- timetable changes (entering through system)
- classes existing in the school (import or entering through system)
- connections between classes, teachers and subjects (import or entering through system)

The administration **shall** have the possibility to add/edit/delete grades.

The login **shall** be generated based on the user name and surname. It **shall** have the following structure:

first letter from user’s name	up to 6 letters from the surname	numerator (if necessary)
----------------------------------	-------------------------------------	--------------------------

#### Ex. mlipink3

There **shall** be no duplications in the login

The password **shall** be generated automatically for all the users whose data is being imported to the system and entered by the administrator if the account is created using the system interface.



### **Third prototyping phase – feedback analysis**

The system **shall** allow to store face-like grades.

The system **shall** allow to store behaviour grades.

The system **shall** allow to store grades according to the term affiliation.

The system **shall** allow to store term grades.

#### **4.3.2 Requirements imposed by clients' software**

The following data concerning pupil **shall** be stored and processed by the system: user login, user password, surname, name, second name, sex, birth date, birth city, PESEL, class name, address, email.

The following data concerning teacher **shall** be stored and processed by the system: user login, user password, surname, name, second name, sex, birth date, PESEL, address, contact number, email, qualification.

The following data concerning parent **shall** be stored and processed by the system: user login, user password, surname, name, second name, sex, PESEL, address, contact number, email.

The following data concerning address **shall** be stored and processed by the system: city, street name, street number, flat number, post code, district, country, kind.

The system **shall** allow to import the following data:

- timetable as a set of html files,
- pupils and parents data from XML file,
- teachers data from Exel file.

#### **4.3.3 Hardware Requirements**

The hardware needed for hosting the system:

- Average class Personal Computer with:
  - Linux or Windows operating system,
  - Apache or other PHP sever, the desired version of PHP is 5,
  - MySQL database,
- Fast internet connection with permanent IP and reasonable upload rate, most preferably separate IP for the computer used as the sever and the school's intranet.

#### **4.4 Optional requirements**

- Information exchange between users (i.e. teacher – parent).
- Ability to send information to a group of users.
- Online school event chronicle.
- Online newspaper with editing tools.

#### 4.5 Requirements prioritisation

*“One characteristic of excellent requirements is that they are explicitly prioritized”*  
**Wieggers, K.(1999)**

All the requirements which were collected before the design and implementation of the first prototype were prioritized in collaboration with the client and presented in a tabular form. Each time new requirements were determined, they were prioritised and added to the table.

The requirement prioritization was done into three priority categories:

- High - a mission critical requirement;
- Medium - supports necessary system operations;
- Low - supports necessary system operations.

The table can be found in **Appendix B: B-5**

#### 4.6 Feedback acquisition

The following techniques were used through the project:

- Survey/questionnaire - is very convenient when the personal meeting is impossible.
- Consultations – with the client as well as with the university colleagues.

Due to the choice of methodology the feedback acquisition was a mean of requirement gathering. As pointed out in the Chapter **4.1 Chosen methodology**, analysis of the feedback can result in requirements change.

The main source of the feedback was the online survey conducted on the second prototype. Feedback was also received from informal online consultations with the client conducted at the end of each prototyping phase.

As the first and second prototypes were put online and updated regularly during the implementation, the client did provide feedback continuously. Since the database was implemented in the third prototype, the prototype could not be put online due to the technical problems. The finished prototype was sent to the client with the appropriate software, sample data and instructions how to install software and run the system (similar to the pack on **CD: POnline\_Pack**). The instruction can be found in **Appendix D**.

The feedback on the first and third prototypes was given by the client and future administrator only during consultation. The main feedback was obtained through the online survey conducted on the second prototype.

#### **A short description of the online survey :**

The survey was conducted for two weeks (25/03/2008 – 7/03/2008). During this time teachers, pupils and parents were encourage to fill up the survey. The most numerous user group to fill the survey were the pupils.

There were three different questionnaires prepared, one for each user group. Each questionnaire consisted of questions appropriate for the group it targeted.

Each questionnaire comprised of three kind of questions:

- General questions, with predefined answers, i.e.
  - “Do you find it difficult to operate the system?”
  - “Would you like to be informed about your child’s new grades? If Yes, should it be done every mark is entered into the system, or you would rather receive the new set of marks at the end of the week?”
- Questions concerning particular pages or functionality of the system. Such questions provided a link to the particular part of the system, so the user could switch to the system, example questions:
  - “Do you think that calculating and showing current average grade of your child and the entire class would be a useful feature?”
- Open answer questions – questions with no predefined answer, i.e.
  - “Please, say in few words what you like about the system and what in your opinion requires changes, what kind of changes?”

To enter the questionnaire the user had to login to the questionnaire page as one of the following users:

- **Login:** pupil, **password:** pupil, **user group:** pupil
- **Login:** teacher, **password:** teacher, **user group:** teacher
- **Login:** parent, **password:** parent, **user group:** parent

Logging to the questionnaire webpage was equal to logging to the POnline system. Thus, user could access the POnline system directly by provided links

The questionnaire can be found here:

<http://home.elka.pw.edu.pl/~mlipins1/questionnaire/logging.php>

A folder with the copy of the system available on line (POnline- prototype 2 and questionnaire) is available on **CD:PSonline\_mion**. The the raw version of the results and detailed analysis of the teachers’ answers (question and answers translated into English) are also provided on **CD:online\_survey\_results**. Figure 4.6-1: shows how each user group answered the following question: “How do you like the system?”  
Screen shot of the questionnaire with description **Appendix B-6**

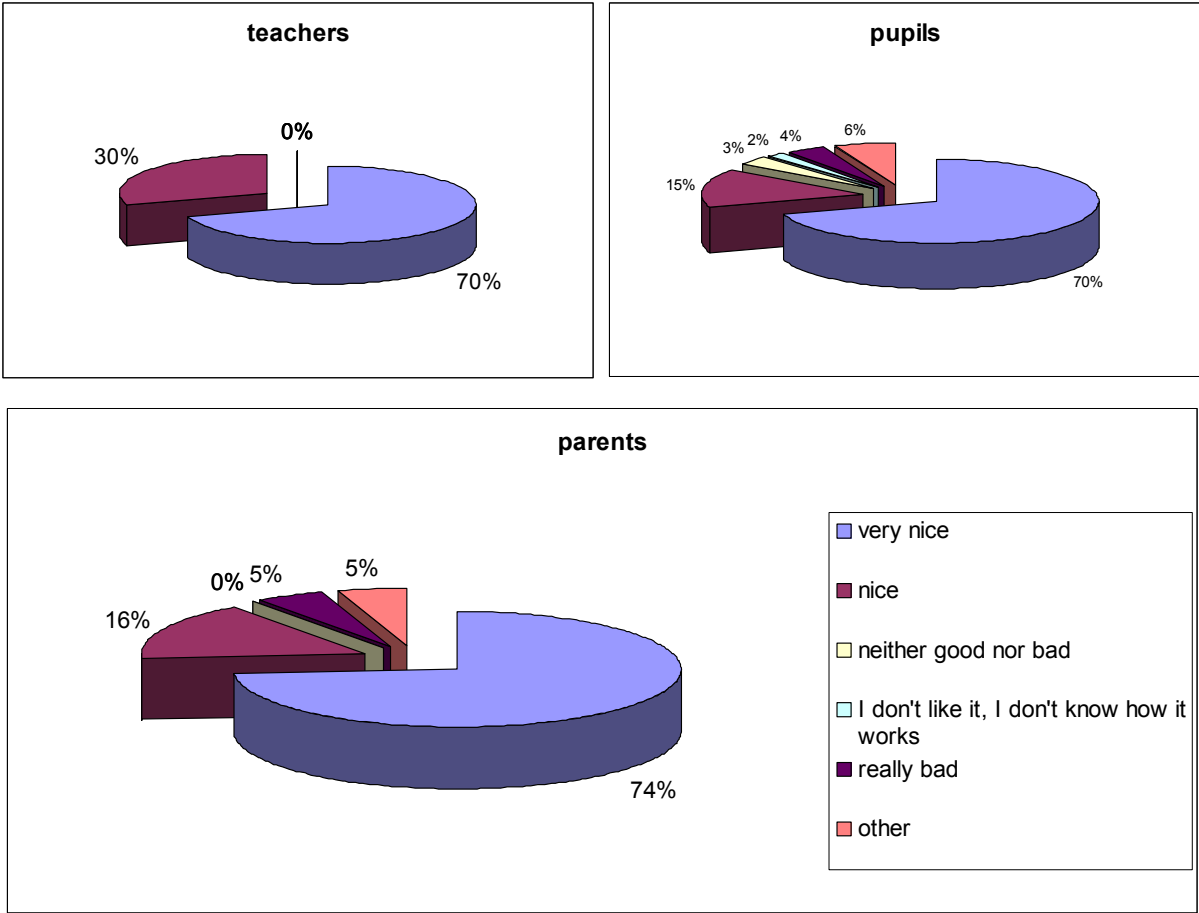


Figure 4.6-1: Online survey outcomes

## 4.7 Analysis

### 4.7.1 Use case diagrams

UML diagrams were used to “describe ‘the meat’ of the actual requirements” and to help in understanding them (**Agile Modeling 2006**).

Four intended user groups were identified. UML allowed to identify a hierarchy of users according to functionality provided to them by the system. The generalization of actors was implemented on the diagrams to show that some characteristics of the system are common to all user groups. The hierarchy of actors is depicted on **Figure 4.7.1-1**.

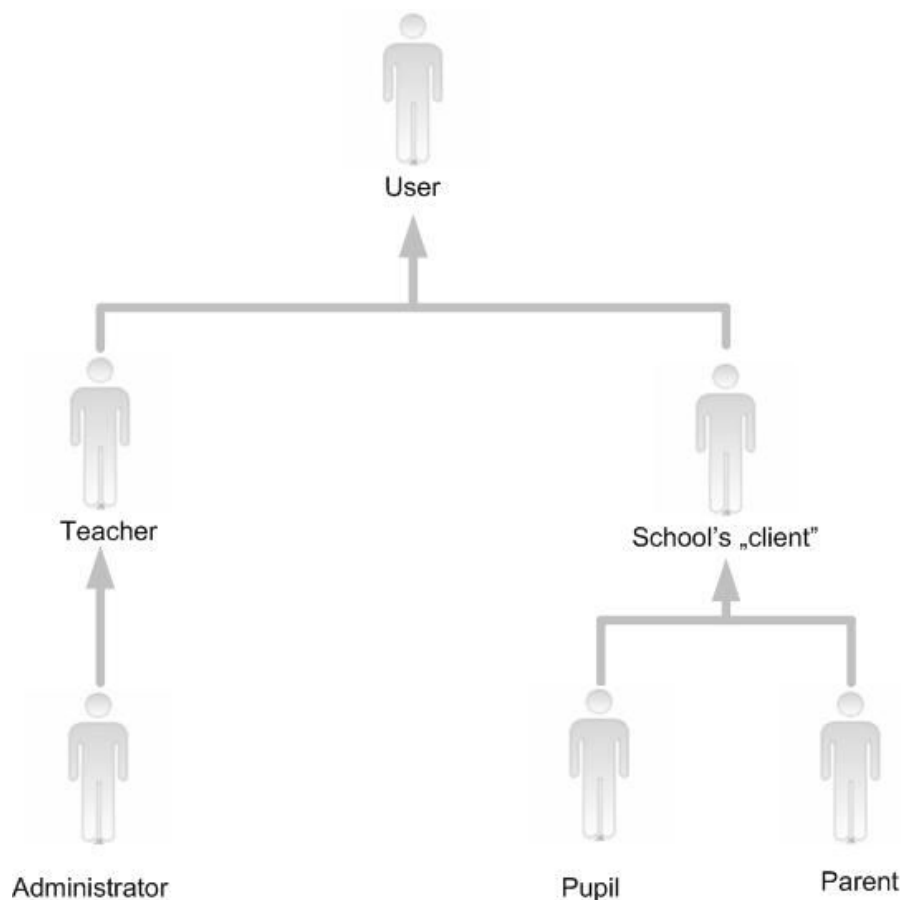


Figure 4.7.1-1 Actors hierarchy

A Use Case Description of the actors can be found along with many other descriptions and Use Case Diagrams on **CD: Requirements/Use Case Diagrams and Descriptions**.

For the clarity of the diagrams only one actor is depicted in the diagrams. All the user groups are generalized by a non-existing actor called “User”. The interactions of the “User” with the system are depicted on **Figure 4.7.1-2**.

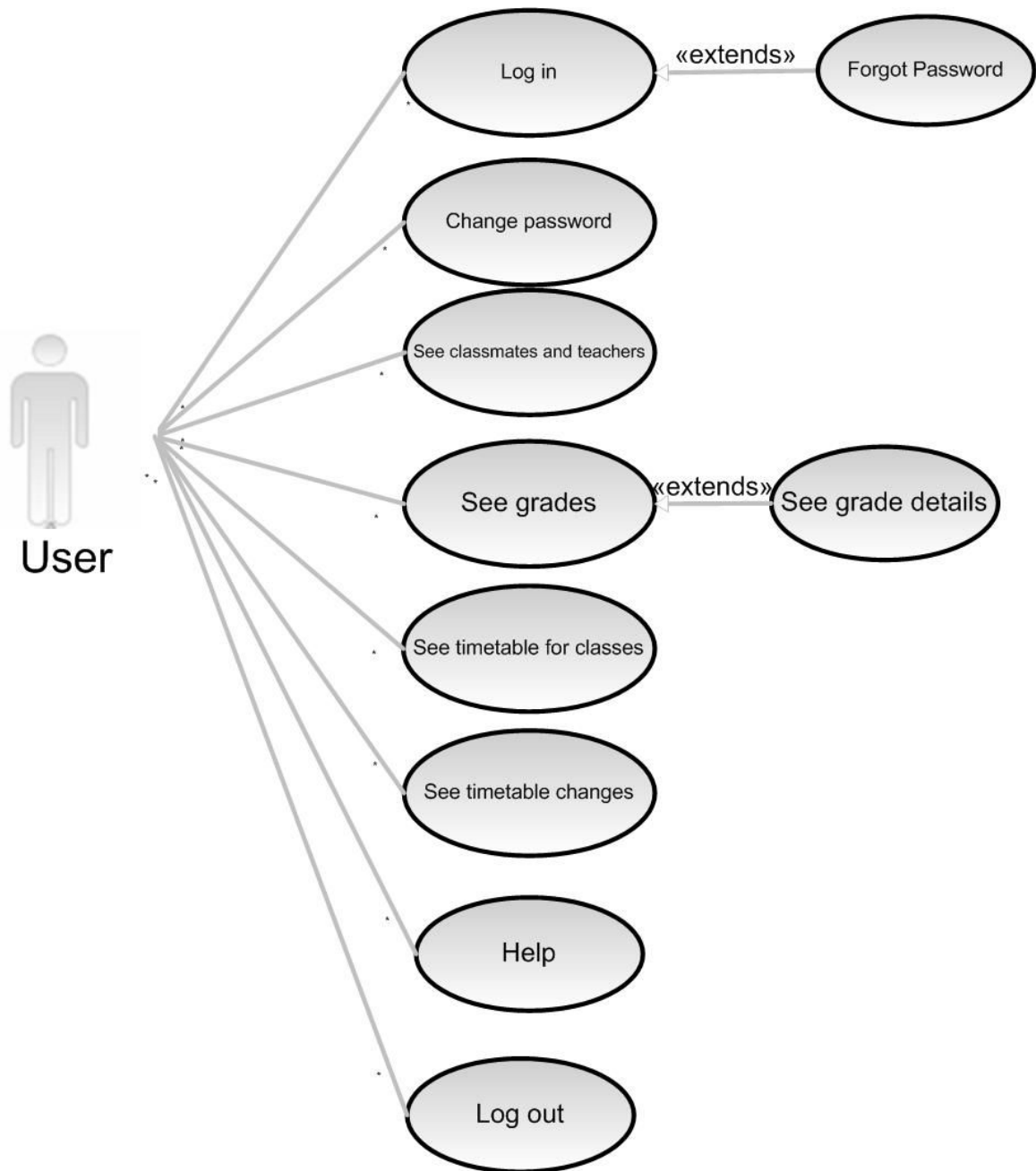


Figure 4.7.1-2 Interactions of a User with the system

**Use cases overview:**

Log in

A user enters the login, password and user group affiliation. If the is correct and the user’s record exist in system database, the user is logged in. If the user has forgotten the password, the user requests the password to be reminded by email.

Change password

User enters the old password and new password and requests the password change.

See classmates and teachers

User requests to see pupils who attend particular class and teachers who teach in the particular class.

See grades

User request to see grades. If needed, user request to see detailed description of a particular grade.

See timetable for classes

User requests to see timetable of a class.

See timetable changes

User requests to see timetable changes for the particular day of the week.

Log out

User requests to log out.

The above Use Case Diagram was extended to include features of the system available for particular actors. Use Case Diagrams with Use Case Overviews seen from each user's point of view can be found on **CD: Requirements/Use Case Diagrams and Descriptions**.

#### **4.7.2 Sequence diagrams**

UML sequence diagrams were used as a bridge between user requirements, presented as Use Case Diagrams and the design. They were used to help translate requirements into design in the case of only few features. In some cases the design was quite obvious and sequence diagrams were not needed.

In **Figure 4.7.2-1** a sequence diagram of the Use Case "Log in" is presented.

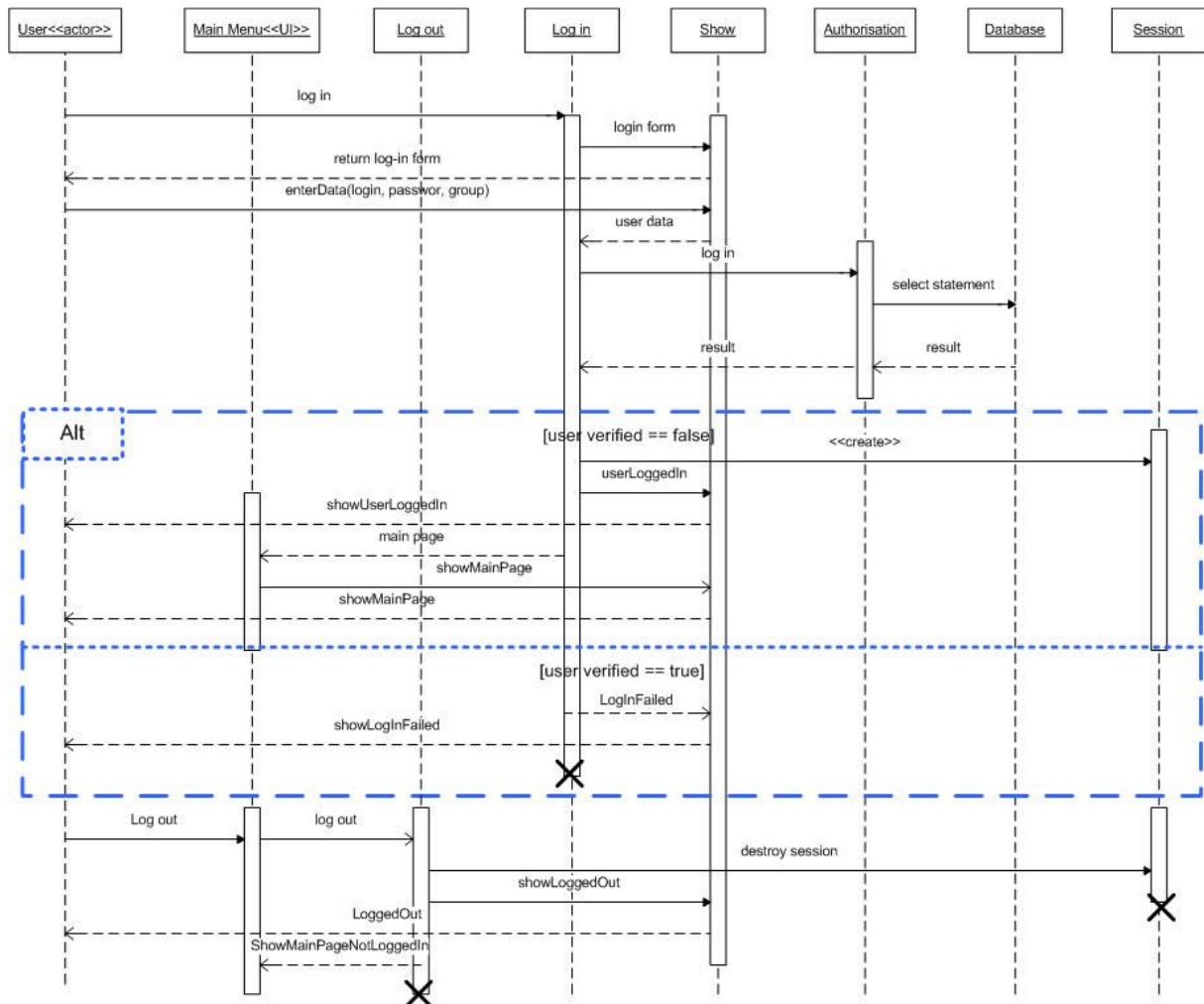


Figure 4.7.2-1 Sequence Diagram of “Log-in” Use Case

Using sequence diagrams was very helpful in showing system’s communication with the database and enabled to establish a well defined database interface without designing its details.

More sequence diagrams can be found on **CD: Requirements/Sequence Diagrams**.



### 4.7.3 Database

Database is one of the most important parts of the designed system. The main task of the system engine is to allow user to see, add and edit data stored in the database.

The process of database analysis was aimed at:

- Establishing requirements affecting database design,
- Point out the information about database which needs to be obtained from the users,
- Produce collection of entities with the attributes.

All the requirements affecting database were collected in one document which is attached on **CD: Requirements/Database Requirements**.

It was established that the database should store the information about:

- Users: administrators, teachers, pupils, parents
- Grades
- Class affiliation by pupils and teacher
- Subject affiliation : subject <-> teacher
- Kinship – affiliation: pupil <-> parent
- Timetable
- Timetable changes

It was decided that the information about each user group would be stored in separate entity– it is justified by the fact that each user group is represented by different set of attributes.

The outcome of the analysis was the following set of entities with desired attributes and primary keys underlined:

**admins** (user\_login, user\_password, surname, name, second\_name)

**adress** (adressID, city, street\_name, street\_number, flat\_number, post\_code, district, country, kind)

**teachers** (user\_login, user\_password, surname, name, second\_name, sex, birth\_date, pesel, adressID, contact\_number, email, qualification)

**classes** (class\_name, leading\_teacher, file\_name)

**pupils** (user\_login, user\_password, surname, name, second\_name, sex, birth\_date, birth\_city, pesel, class\_name, adressID, email)

**parents** (user\_login, user\_password, surname, name, second\_name, sex, pesel, adressID, contact\_number, email)

**subjects** (subjectID, subject\_name, subject\_name\_eng)

**grade\_kinds** (kindID, kind\_name, kind\_name\_eng)

**grade\_descriptions** (descriptionID, description)

**pupil\_parents** (pupil\_parentID, pupil\_login, parent\_login, kinship, legal\_guardian)

**class\_subjects** (class\_subjectID, subjectID, class\_name, user\_login)

**grades** (gradeID, user\_login, class\_subjectID, grade, date, kindID, descriptionID)

**f\_places** (lf\_placeID, place\_name, place\_name\_eng)

**corridor\_lfs** (corridor\_lfID, user\_login, lf\_placeID, lf\_time, day\_name)

The database needed for the system does not seem to be complicated and its size is limited by the size of the school. The maximum number of each table records can be roughly estimated. Currently the number of pupils attending the school is approx 230, the number of parents is approx  $2 \times 230 = 460$ . The number of teachers is approx 30. Assuming that the above numbers can double in the worse case and that the table which will possibly contain the greatest number of rows, namely “grades”, stores only very simple data (characters and digits), the choice of relational model database (Chapter: **3.5. Technology review, competition and justification**) seems sensible and sufficient for this project.

## 5. Design

### 5.1 System architecture

*“A system's architecture represents the necessary strategic design decisions sufficient to form that system.” (Booch, G.2001)*

The system architecture of PSONline is depicted in **Figure 5.1-1**. This is a client-server standard architecture of systems based on PHP scripting language and SQL database. The architecture is determined by the technology chosen for project implementation, namely WAMP/LAMP (Windows/Linux, Apache, MySQL, PHP). The client in **Figure 5.1-1** is represented by the Web Browser and the LAMP/WAMP server consists of :

- Web server – HTTP server which accepts HTTP requests from the client and sends responses,
- PHP script-processing module – creates HTTP responses to be sent by web server,
- File system – stores scripts, HTTP pages and other data,
- Database – stores data.

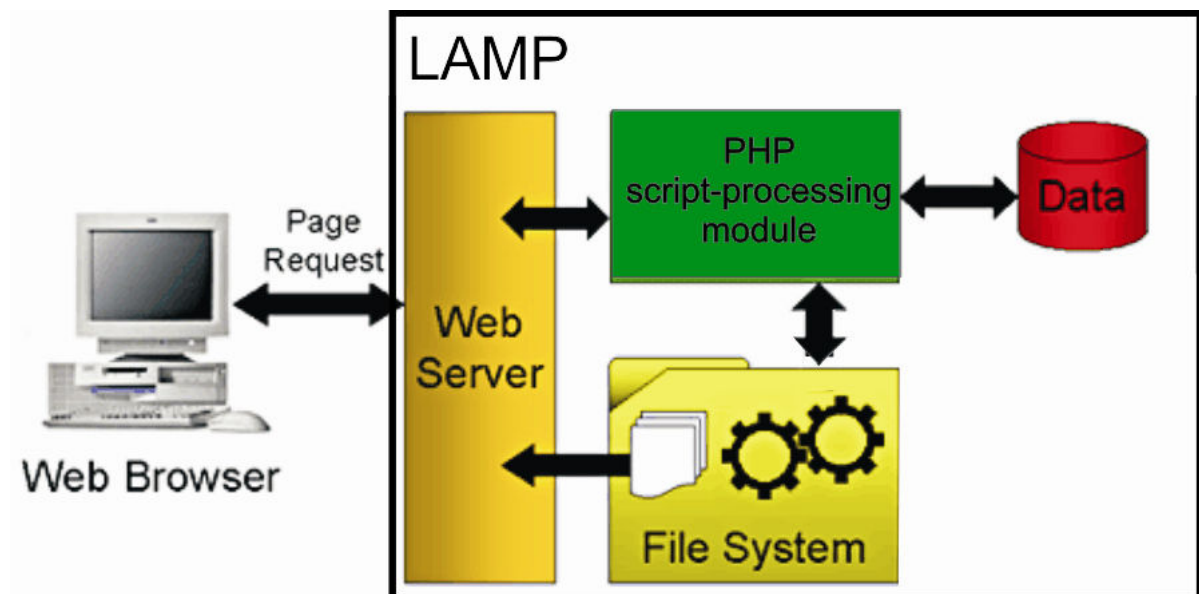


Figure 5.1-1 System architecture (Booch, G. 2001)

## 5.2 Software architecture

Three kind of client-server approach can be identified, namely two-tier with thin client, two-tier with fat clients and three-tier/multiple architectures. To establish the best solution for POnline design a table listing the best application for each architecture from **Sommerville, I.(2007: 275)** was used.

The decision to choose three-tier architecture was base on the following characteristics of the system:

- Considerable number of clients: pupils + parents + teachers = 230 + 460 + 30 = 720,
- Data entered by multiple clients (teachers, administrator),
- Data updated daily,

Three-tier architecture consists of three independent modules/tier Sommerville, L. (2007: 270):

- Presentation tier - Client interface – presenting information to the user and with all user interaction,
- Application processing tier– business logic –implementing the logic of the application,
- Data management tier - Computer data storage –all data operations

Thanks to modularity, any of the tiers can be upgraded or changed without having to modify the other tiers.

Based on the chosen architecture pattern and the requirements the architecture of POnline was established (**Figure 5.2-1**).

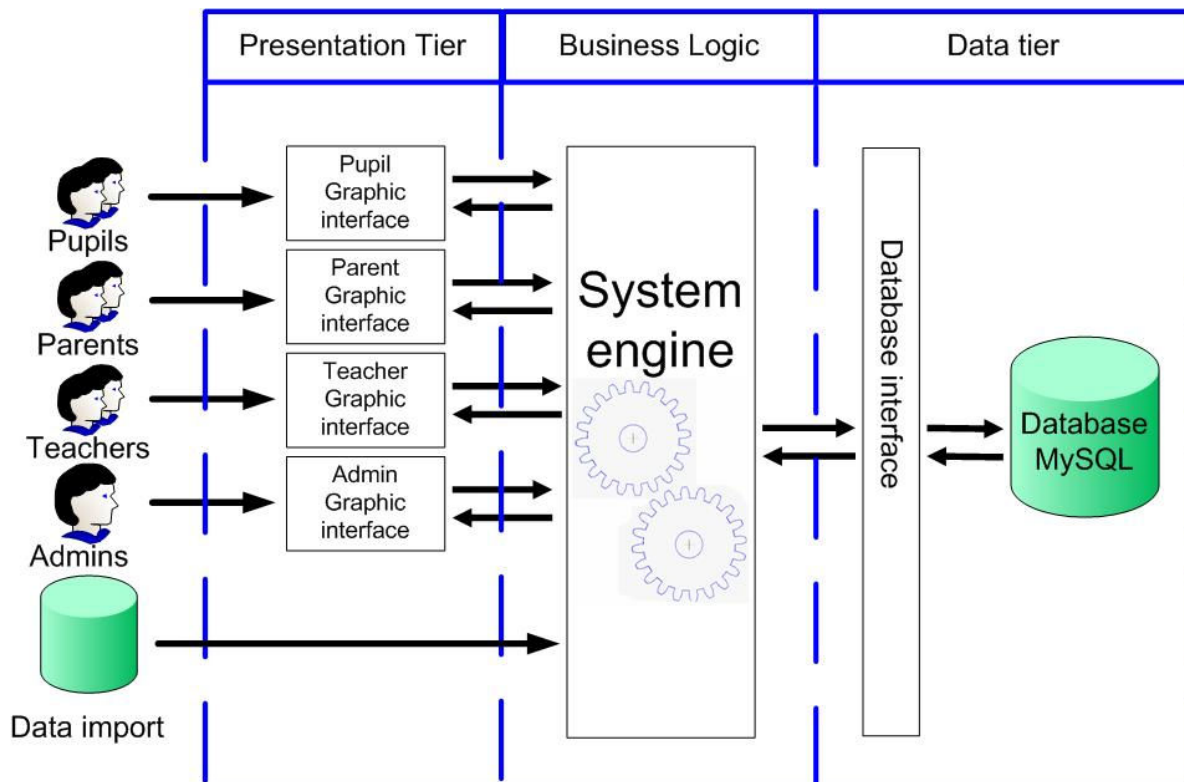


Figure 5.2-1 POnline architecture

### 5.3 Database design

Although the database implementation was planned for the development of the third prototype, a logical model of database was created in the first phase. It would be impossible to design and implement first two prototypes without database design.

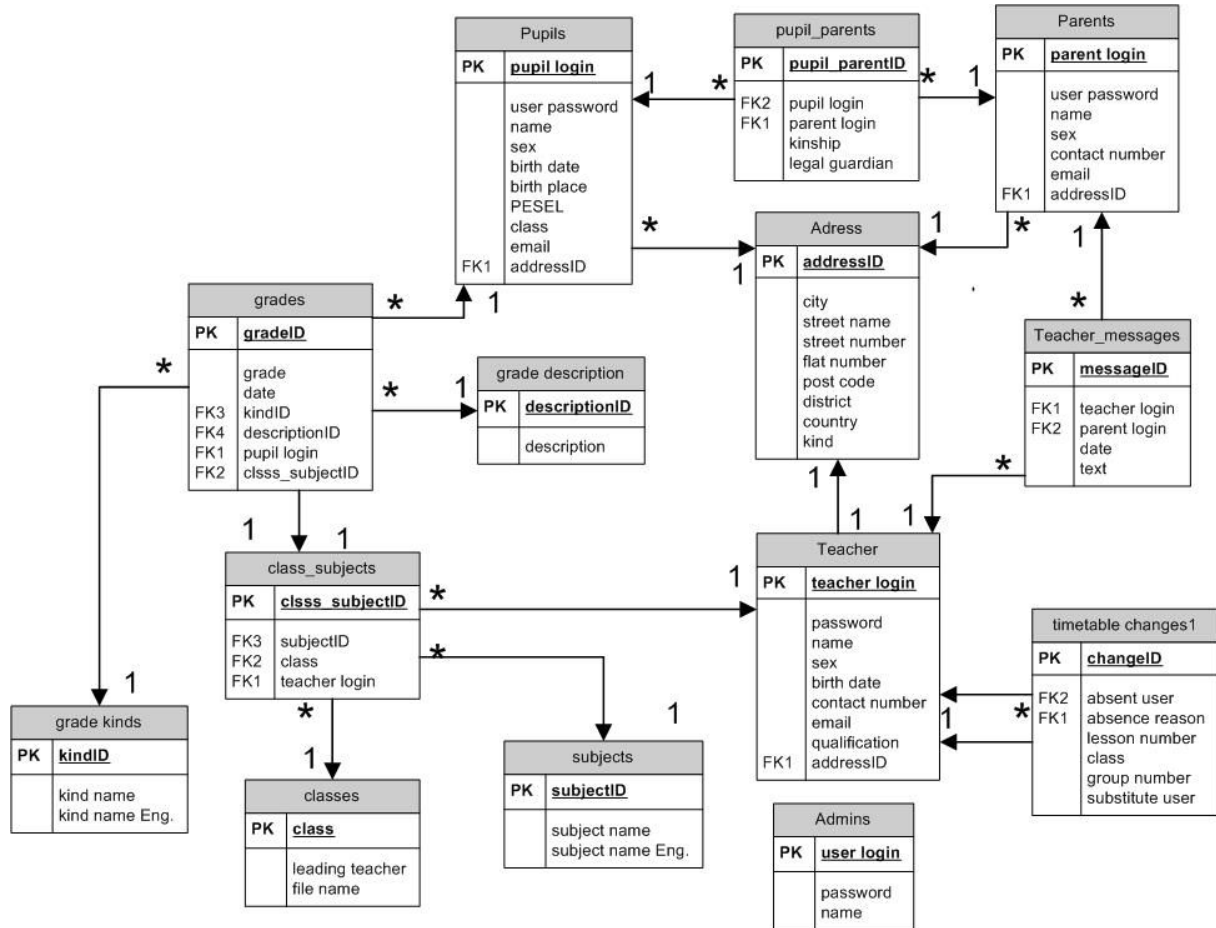


Figure 5.3-1 First database design

**Figure 5.3-1** depicts the logical database model based on the analysis of database performed in Chapter 4.7.3 Database.

When designing logical model of relational database all the necessary rules were obeyed and Normalization Forms used (Litwin, P. 1990 ; Silberschatz, A. 2001).

#### Modifications to the database design based on user feedback

In the design of the third prototype modifications to the initial database model were introduced to include the user feedback. The online survey pointed out that in levels 1-3 descriptive grades are used. I was also discovered that term grades and year-grades need to be implemented as well as the distinction between two terms. In the feedback after the third prototype it was discovered that graphic grades and behaviour grades need to be implemented. What is more, mistakes were discovered



## 5.4 Prototypes design

Due to the methodology choice the design of prototypes is important from the very beginning of the project. Each prototype must be designed in such a manner that it can be easily changed to meet users' requirement collected in the feedback and be a good basis for further development of the project.

**Table 5.4-1** shows which features were planned to be designed in each development phase.

The design of the first prototype produced the basis for entire project. Deliberately, it was decided to design the functionalities and interface for pupils in the first place. From the Use Cases in the Chapter **4.7.1. Use Case diagrams** it is clear that pupil's features are inherited by all the users. Consequently pupil's user interface constitute basis for all other user group interfaces.

The design of the project is based on the book **Welling, L. & Thomson, L.(2005)**. The book gives a great insight into PHP web application and provides useful examples. The book did not encouraged to use object-orientation in PHP. All of the examples were not created as object-orientated. Therefore the decision was made to design and create the system in structural manner. However, the way the functions are divided into the files makes the system modular. The file organization is described in the Chapter **6. File organization**.

phase	What was designed during the prototyping phase	Design vs. architecture
prototype one	<ul style="list-style-type: none"> <li>• engine with appropriate features available for pupils user group,                             <ul style="list-style-type: none"> <li>○ browsing classmates and teachers</li> <li>○ browsing grades</li> <li>○ browsing timetable</li> <li>○ browsing timetable changes</li> <li>○ changing password</li> </ul> </li> <li>• pupils' interface</li> <li>• scratches of database interface</li> </ul>	
prototype two	<ul style="list-style-type: none"> <li>• features of the system appropriate to parents:                             <ul style="list-style-type: none"> <li>○ choosing child to see the information about, if more than one available</li> </ul> </li> <li>• features of the system appropriate to teachers:                             <ul style="list-style-type: none"> <li>○ browsing detailed information about other users</li> <li>○ adding, editing, and deleting grades</li> <li>○ browsing timetable changes for teachers</li> </ul> </li> <li>• teacher's and parents interfaces</li> <li>• scratches of database interface</li> </ul>	
prototype three	<ul style="list-style-type: none"> <li>• functionalities of the engine specific to administrator                             <ul style="list-style-type: none"> <li>○ adding/editing/deleting all users</li> <li>○ adding/deleting timetable changes</li> <li>○ adding/editing/deleting classes</li> <li>○ importing data</li> <li>○ browsing all users</li> </ul> </li> <li>• administrator interface</li> <li>• database and complete database interface</li> </ul>	

Table 5.4-1 Prototyping phase of the design



### 5.4.1 Prototype 1 design

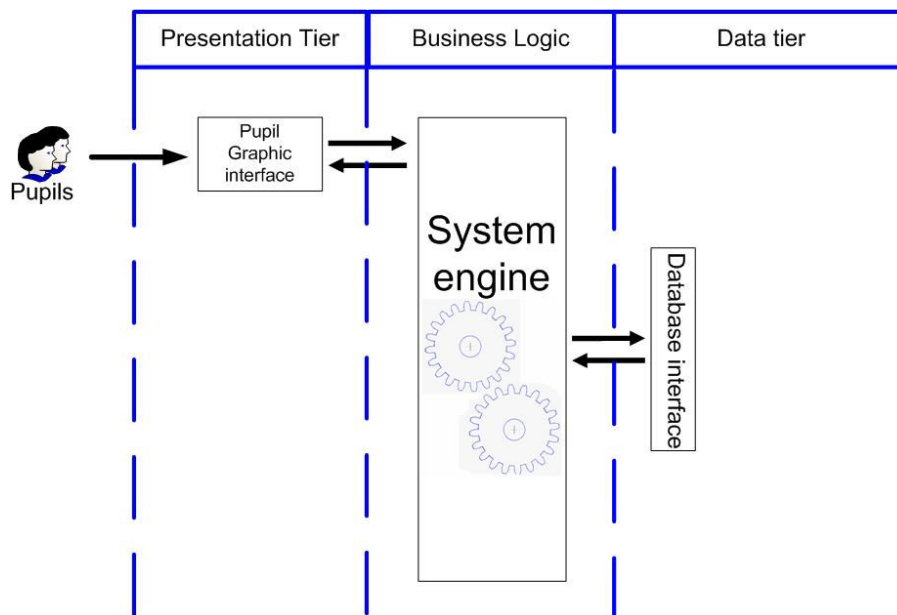


Figure 5.4.1-1 Prototype architecture

**Figure 5.4.1-2** depicts Activity Diagram of the prototype one. The diagram is straightforward and clearly shows that the user interface is not complicated. The only sophisticated service is “see grades” which enables pupils to see their grades.

Establishing the Activity Diagram was very important to design the layout of the webpage. The information displayed on the webpage can be divided into:

- information that is always displayed, namely:
  - heading:
    - school name and logo
    - user name and login
    - menu
  - footstep
    - menu
- information that depends on the system state
  - classmate and teachers
  - timetable
  - timetable changes
  - password change form

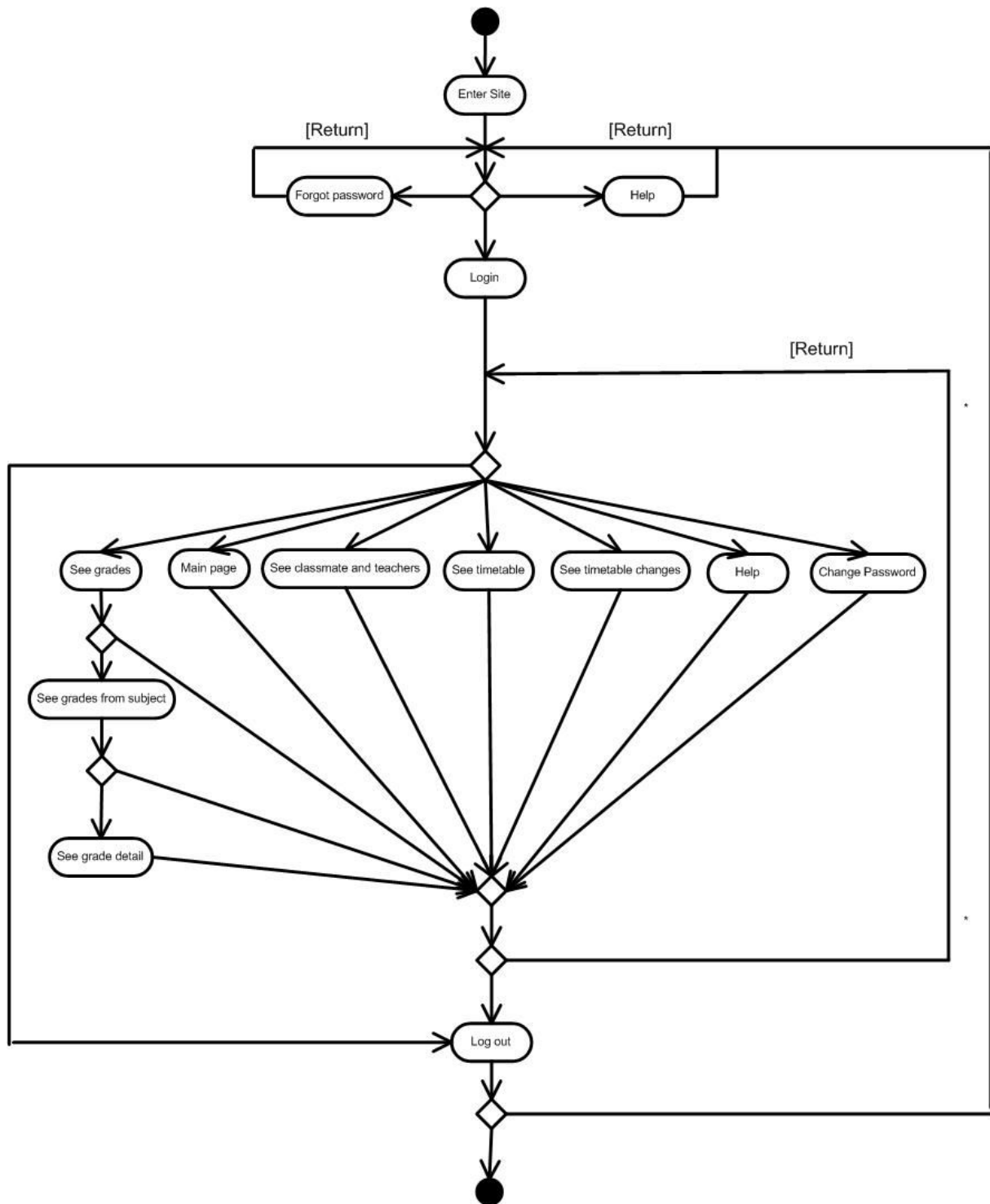


Figure 5.4.1-2 Activity Diagram of pupils interface

The following layout of the webpage was accepted by the client for implementation in the first prototype.

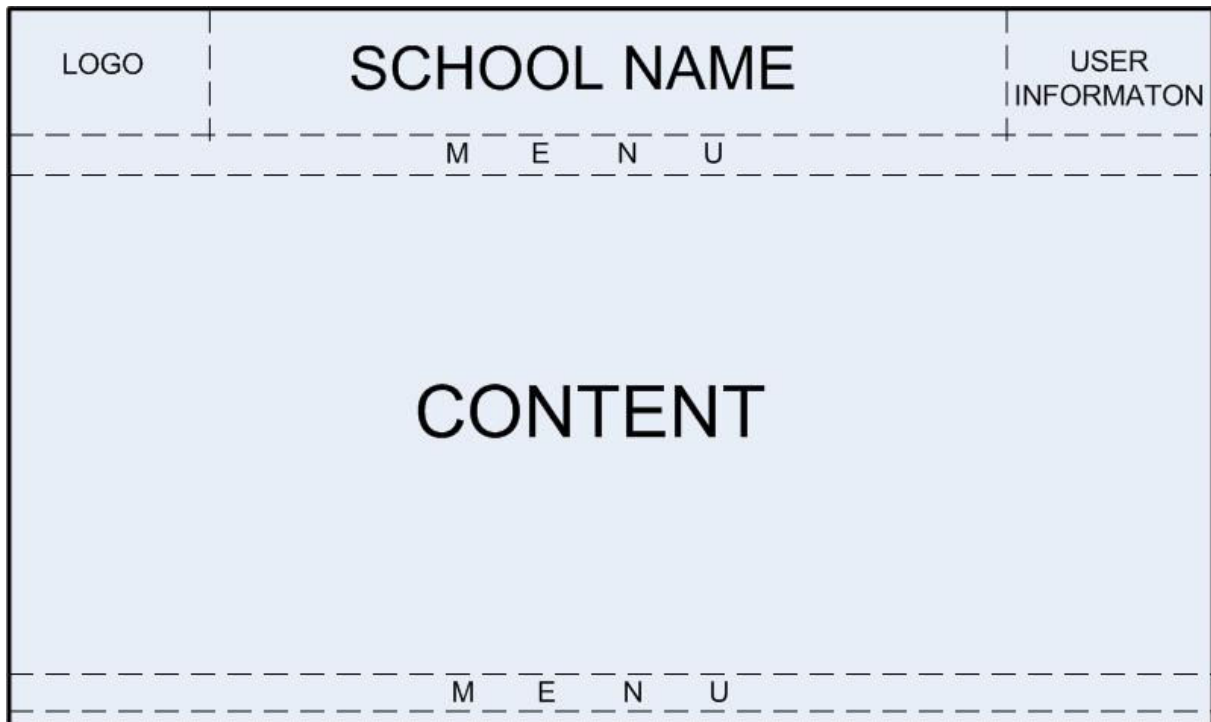


Figure 5.4.1-3 Webpage layout

The way the grades should be displayed had to be discussed during consultations with the client. The following considerations needed to be taken into account when displaying grades:

- One pupil has grades from many subjects;
- Grades from one subjects have different kinds;
- There are four predefined kinds : test, activity, homework, others, however the requirement states that teacher can add new kinds if the four predefined kinds are not sufficient;
- On one day more than one grade can be given to one pupil from one subject.

It was decided that hierarchical approach is the easiest way to show information about the grades in a clear and intuitive manner. The idea of displaying grades with the range of information increasing and the number of information decreasing when entering higher levels of hierarchy (**Figure 5.4.1-4**).

- First level - general view - grades from all subjects divided onto four predefined kinds, if new kinds shown in the column "others". Additionally, grade average displayed:
  - Average of the grades from one kind;
  - Average from all pupil's grades;
  - Class average.
- Second level - subject view – grades from one subject - the grades displayed ordered by date and divided into all the categories that they fall into.
- Third level – grade detailed information – all the information about one grade displayed.

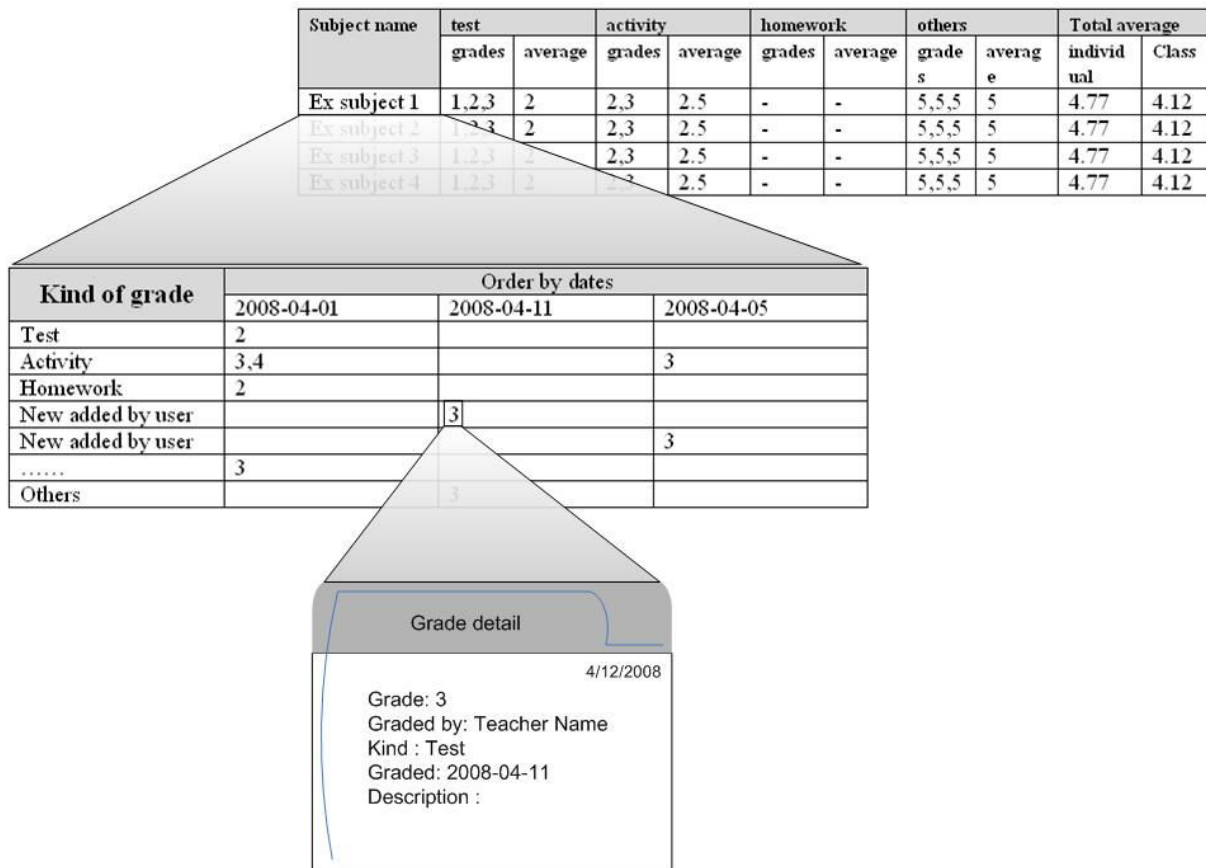


Figure 5.4-1.4 Hierarchical grade display

**Figure 5.4.1-5** presents an Activity Diagram depicting the process in which the website displaying information about the grades is created. This is an example which illustrates a general rule. Each webpage which shows information retrieved from database would have very similar Activity Diagram. The diagram shows the division between three tiers constituting the system architecture and how the tiers communicate and interact.

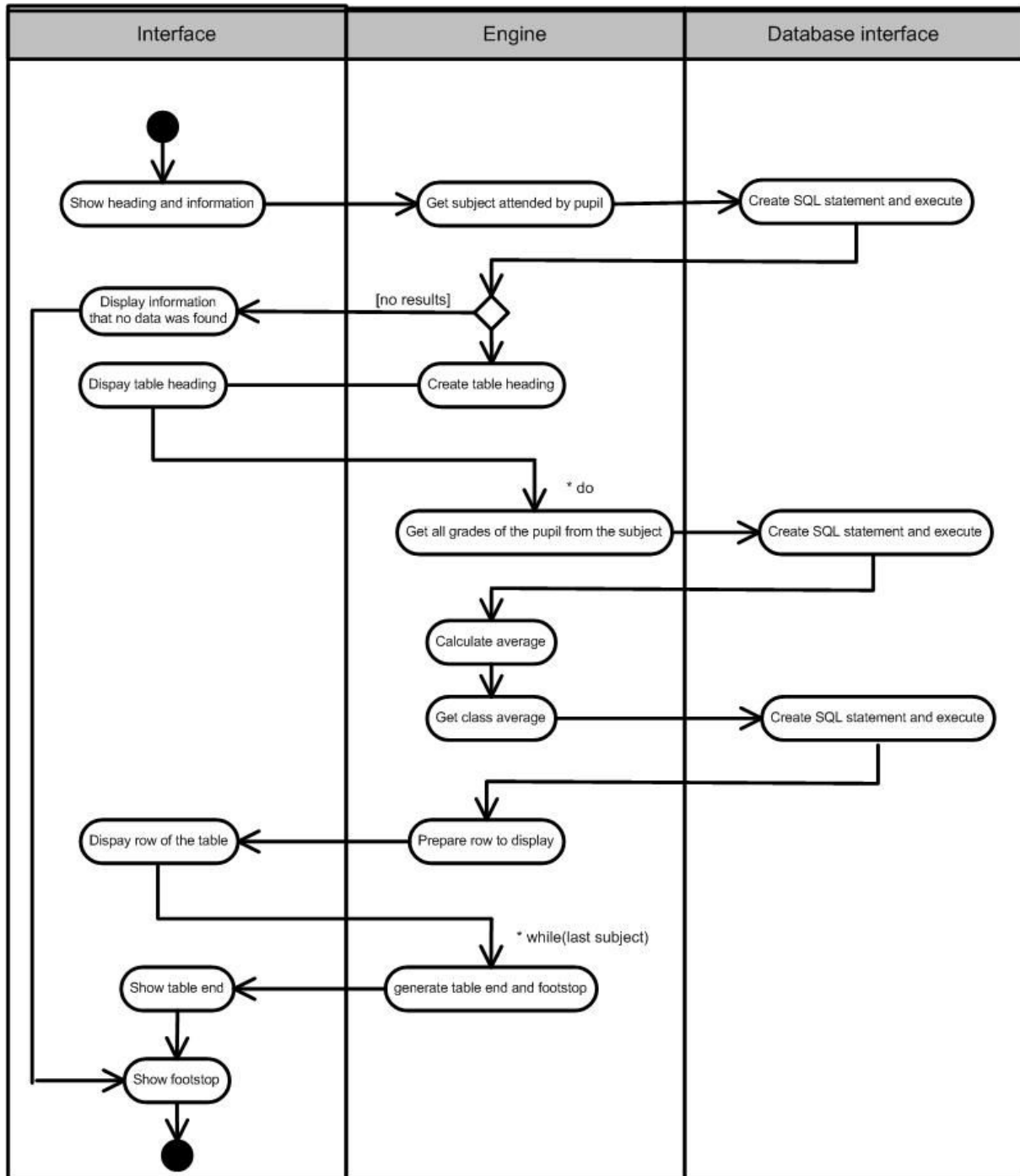


Figure 5.4.1-5 Activity Diagram

The design of database interface depicted in the **Figure 5.4.1-5** is limited to defining functions which would normally retrieve the data from the database and establishing what data should be returned by the function.

### 5.4.2 Prototype 2 design

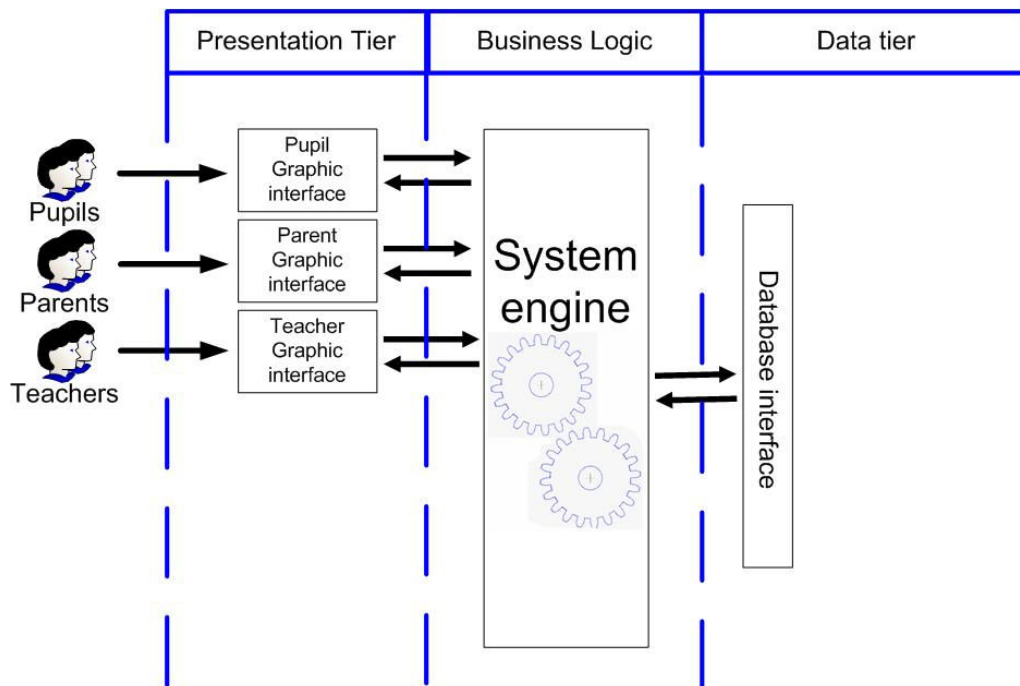


Figure 5.4.2-1 Prototype architecture

Designing the mechanisms and interface of editing, adding and deleting grades and redesigning the system to include user feedback concerning the language requirement were the biggest challenges in this phase.

**Figure 5.4.2-2** depicts Activity Diagram of grades management in the teacher interface. Implemented in pupil interface hierarchical approach of displaying information about the grades was adapted:

- The first level view of grades - all grades for all the pupils in a class taught by the teacher and the options available:
  - see pupil's details,
  - see all the grades received by the pupil ordered by date and kind ,
  - see all the grades received by the pupil ordered by date and kind in order to edit particular grade,
  - add new grade,
- The second level view - the same as in pupils interface (whether the grades are to be edited or not),
- The third level few - depending on the option chosen in first level view –
  - details of the subject which cannot be altered ,
  - details of the subject which can be altered/deleted.

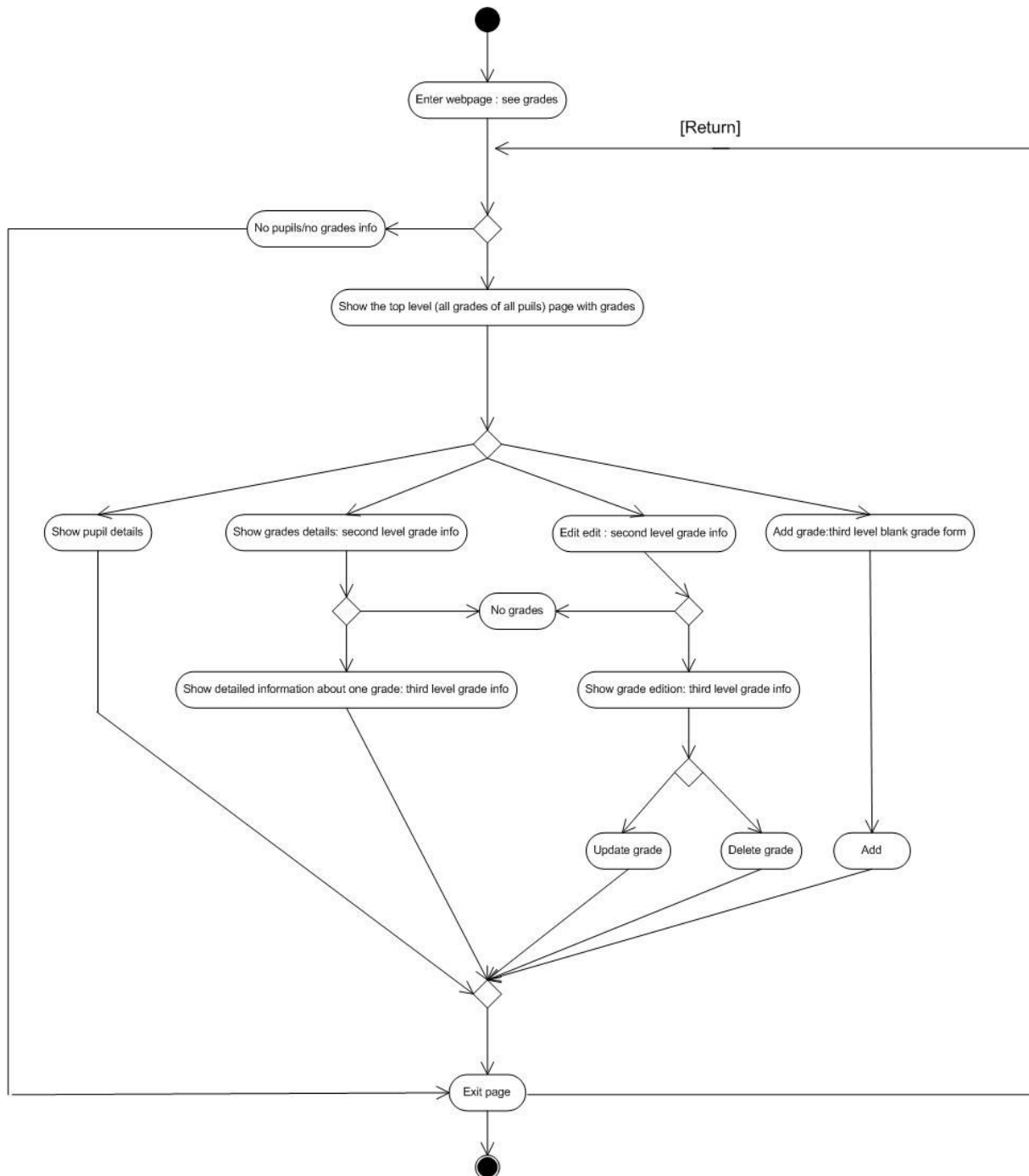


Figure 5.4.2-2 Activity Diagram – “see grades”

The feedback of the first prototype revealed that the client is not satisfied with the product to be operated in English. Thus the first design had to be altered to meet a crucial requirement: **the system needs to be operated in Polish**. However, as the basis had been already created in English, deleting English instructions and interface information would be a waist of time and work. Therefore it was decided to redesign the system to allow bilingualism.

### 5.4.3 Prototype 3 design

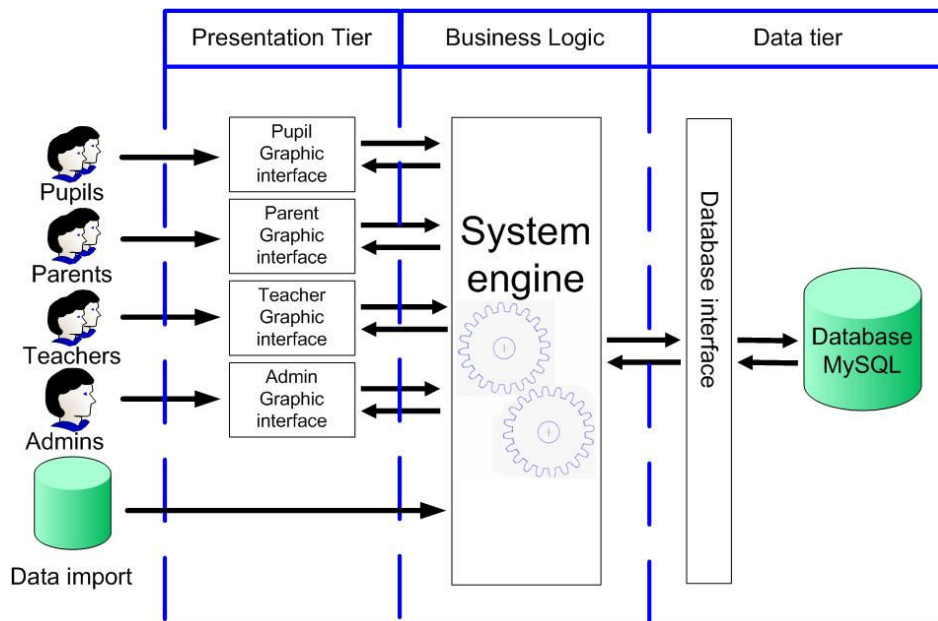


Figure 5.5.3-1 Prototype architecture

The design was started with making the following changes to the previous design due to the feedback:

- Descriptive grades in levels 1-3,
- Possibility of adding grades to many pupils at the same time

Due to the feedback collected the preliminary database design had to be changed. Database design and its evolution is described in the Chapter **5.3 Database design**.

The administrator part of the system is based on the teacher interface and engine features. It extends these features by a considerable number of database-related (editing, adding, deleting) operations and data import. Figure **5.5.3-2** shows the Activity Diagram of the service providing information about classes (pupils and teachers connected with a particular class) and enabling administrator to edit classes, pupils and teachers.



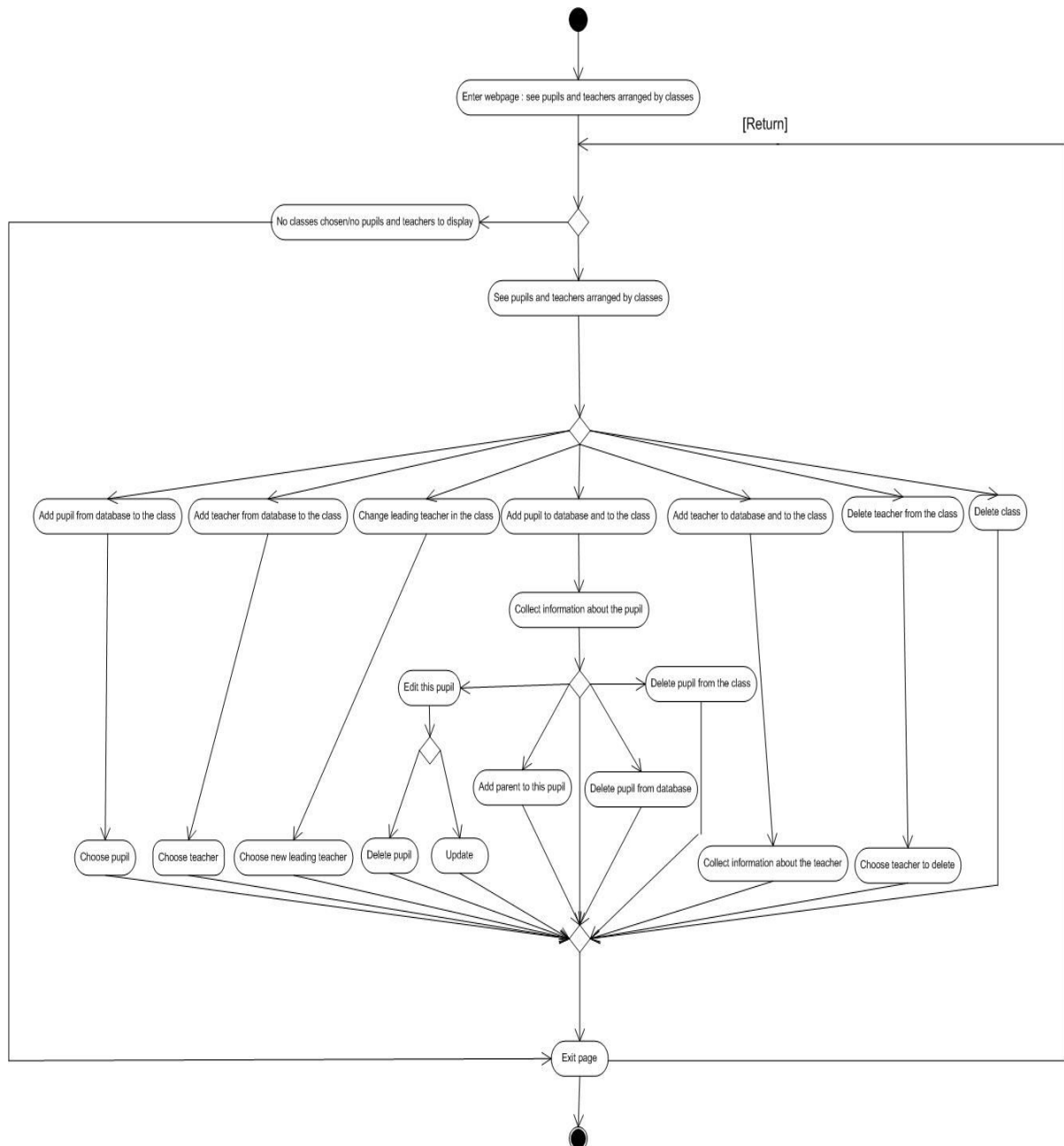


Figure 5.5.3-2 Activity Diagram – “see pupils and teachers arranged by classes”

The following, additional to teachers’ design, features had to be included in the administration design:

- Browse all users of the given user group,
- Import data,
- Delete previous database.

It was decided to create a user called “superadmin” - the emergency administrator’s account which cannot be deleted and password cannot be changed. It would be created to prevent the situation when all the administrators are deleted, which is possible as the administrator has unlimited right to do changes in the database. The password to this account will not be revealed to the client, unless the emergency situation arises.

More Activity Diagrams can be found on **CD: Design/Activity Diagrams**

The approach to data import is following:

- Upload and save the import file,
- Read the file,
- Insert the read information into the database,
- Delete the file.

## **5.5 Final design**

The feedback on the third prototype revealed that the following changes were needed to fulfil users' requirements and needs:

- Term grades and division into two terms of the depicted grades,
- Face-like grades in levels 1-3, which are used together with normal grades,
- Behaviour grades.

The changes required redesign of the database which is described in the Chapter: **5.3 Database design**. The way the grades were displayed had to be changed in levels 1-3 to meet clients requirements.

The administrator interface for data import was extended. A mechanism of enabling and disabling import of particular data depending on the state of other data was designed to enforce the right sequence of data import. It is because the tests on prototype 3 proved the right order of data import to be crucial.

## **6. Implementation**

Through the implementation many casual tests were held to prove that the implemented part of the system works before stating that the prototype is completed. However the tests were not documented and are not described in the Final Report.

### **6.1 *Hardware used during the development***

First and second prototypes were ran on the developers computer (PHP 5, MySQL) and on the server provided by The Faculty of Electronics and Information Technology of Warsaw University of Technology. The faculty's server was used to conduct the online survey and run the prototype system tested by intended users for the feedback. The faculty's server provides PHP (version 4). The server does not provide MySQL database, however the first and second prototype did not use database.

Third prototype and the final product of the project were ran on developer's computer and on the client's computer only.

### **6.2 *File organization***

PHP is a computer scripting language. A PHP web application consists of many files with PHP and HTML code. To make the design and development easier a file organization policy was adapted. It is base on the file organization policy used in the provided examples to Welling, L. & Thomson, L.(2005).

File organization and naming rules are determined by three-tier architecture and by web-site navigation. It is explained in more details in the Table 6.2-1.

FILE NAME OR FILES NAMES	CONTENT	T I E R
Texts.php	All the texts with instructions or information for the user are placed in one file. It allows the client to make necessary changes by himself	I N T E R
Show_functions.php	All the functions which are responsible for visualisation, including HTML code etc	R F A C E
Database_functions.php	Functions responsible for connecting with database	D A
Data_mining_functions.php	Functions creating desired SQL queries	T A
Authorization_functions.php	Logging in and verifying the user	
Data_import_functions .php	Functions which perform file upload and manage reading data from the files (XML, xls) to the database	B U S
“user_action”.php Where user_action represents the service a user receives from the system, ex. delete_grade.php All the services that are available only for the administrator have a prefix “admin”	Represents all the services that the system enables to the users.	I N E S S

Table 6.2-1

All the functions responsible for displaying information are grouped in one file “text.php”. Such organization makes changes to the information displayed in the system very easy and possible for a non-programmer, such as the client. A special way of naming the functions responsible for showing text information was established. The name (ex. multiple\_grade\_detail\_info() ) consists of :

- File name in which the function is used, ex: grade\_detail.php,
- Additional information if necessary ex. multiple,
- Suffix: info,

This system was created to allow the client to correct or add new information by his own. The client, when seeing a mistake or wrong information displayed in the system, checks the url address for the file name. File names always end with “php” suffix so it is fairly easy to recognize. Once the name has been established, a search has to be done in the file “text.php” for the function called filename\_info() or similarly. It was used to make all necessary corrections in the texts and information displayed in the system and can be used in the future.

### 6.3 Database implementation

Database implementation consisted of two phases:

- Creating data base in MySQL,
- Implementing the database interface.

**Figure 6.3-1** depicts the set of commands used to create database and four users with varied privileges. It is very important that a user owns the lowest number of privileges possible so the changes in database are not performed by unauthorized user. In the real implementation the passwords have to be agreed.

```
1 create database psonline;
2 use psonline
3
4 grant select, delete, insert, update
5 on psonline.*
6 to teacher@localhost identified by 'psonline_teacher';
7
8 grant select
9 on psonline.*
10 to pupil@localhost identified by 'psonline_pupil';
11
12 grant select
13 on psonline.*
14 to parent@localhost identified by 'psonline_parent';
15
16 grant all
17 on psonline.*
18 to admin@localhost identified by 'psonline_admin';
```

Figure 6.3-1 Example commands creating POnline database

To make working with database easy a set of commands creating tables and inserting initial sample data was saved in SQL file (**CD:implementation/sample\_database**) and imported to the database by executing appropriate command.

#### Integrity rules

The integrity of data in database is check by the database itself with the mean of foreign key referencing and using the InnoDB engine.

## 6.4 Prototypes implementation

### 6.4.1 Prototype one

The implementation of prototype one was started with creation of log-in and authorisation mechanisms. It is the very important part of the system and therefore needed to be implemented and tested with a great care. The Sequence Diagram presenting the log-in logic is depicted in **Figure 4.7.2-1** in the Chapter 4.7.2 **Sequence Diagrams**. Logging mechanism is the only one in which database interface was not limited to hard coding the data into database interface function but the data was stored in the file located out of Root File System to ensure at least a small level of security.

**Figure 6.4.1-1** shows how the design was implemented. In an object-orientated design, a sequence diagram is used to define classes and functions. Although POnline implementation is not object-orientated, the sequence diagram was used to divide in a logical manner system functions into files. For the simplicity sake it is assumed in this diagram that the user enters the right data and no alternative actions are depicted.

A screen shot of the log-in page is attached in **Appendix C:C-1**, the login form is shown in **Figure 6.4.1-2**. First prototype was developed in English.

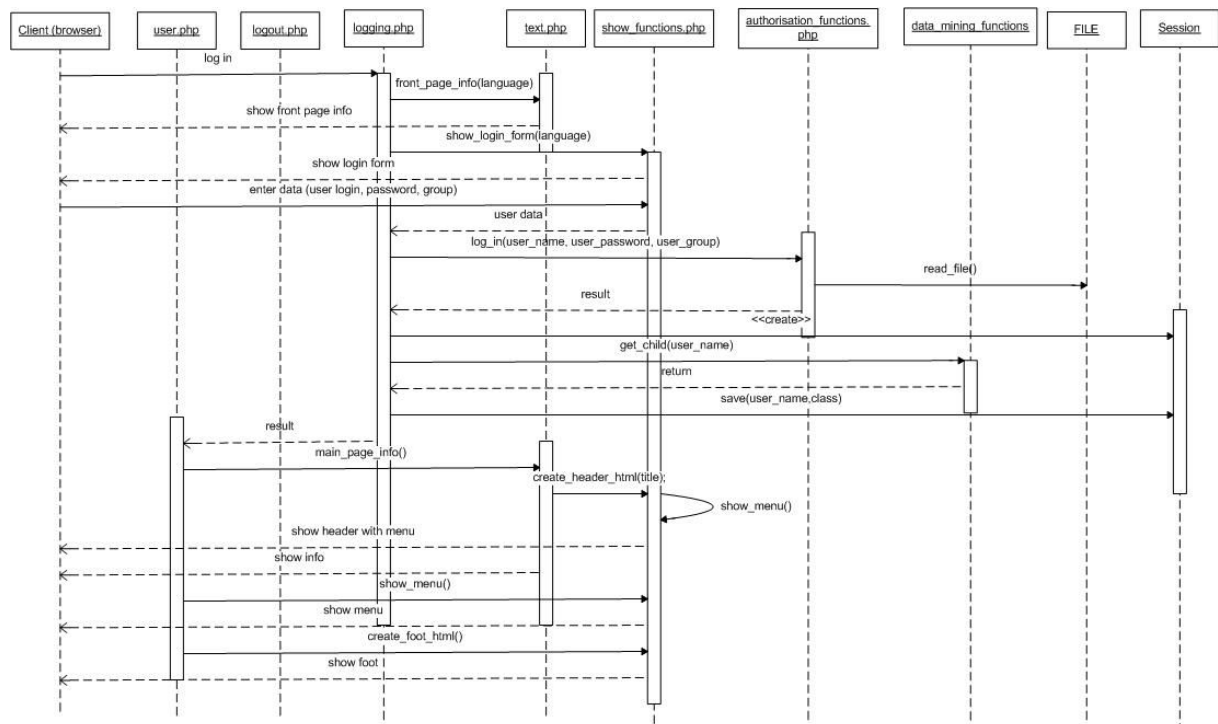


Figure 6.4.1-1 Implementation of Sequence Diagram

Figure 6.4.1-2 Login form

The implementation of website layout was meant to be simple in use. Webpage layout is always created with two functions:

- **Create\_header\_html()** – displays the header: school photo, school name, system name, menu and information about user who is logged in,
- **Create\_foot\_html()** – displays the foot: bottom menu.

As an example a code of the file responsible for displaying the log-in website is depicted in **Figure 6.4.1-3**.

```

1 <?php
2     require_once ('functions.php');
3     create_header_html ('');
4     logging_info ();
5     show_login_form ();
6     create_foot_html ();
7 ?>

```

Figure 6.4.1-3 – Log-in webpage PHP code

Firstly, all the files containing necessary functions are included by **require\_once()** function, then the function responsible for displaying header is called. Function **logging\_info()** is responsible for displaying text information. Next function is responsible for displaying the log-in form. **Logging\_info()** and **show\_login\_form()** are the functions responsible for displaying website content. The last function creates a website foot which shows the bottom menu.

Below, in Figure 6.4.1-4 a simplified (only two levels deep) site diagram of pupil interface is presented.

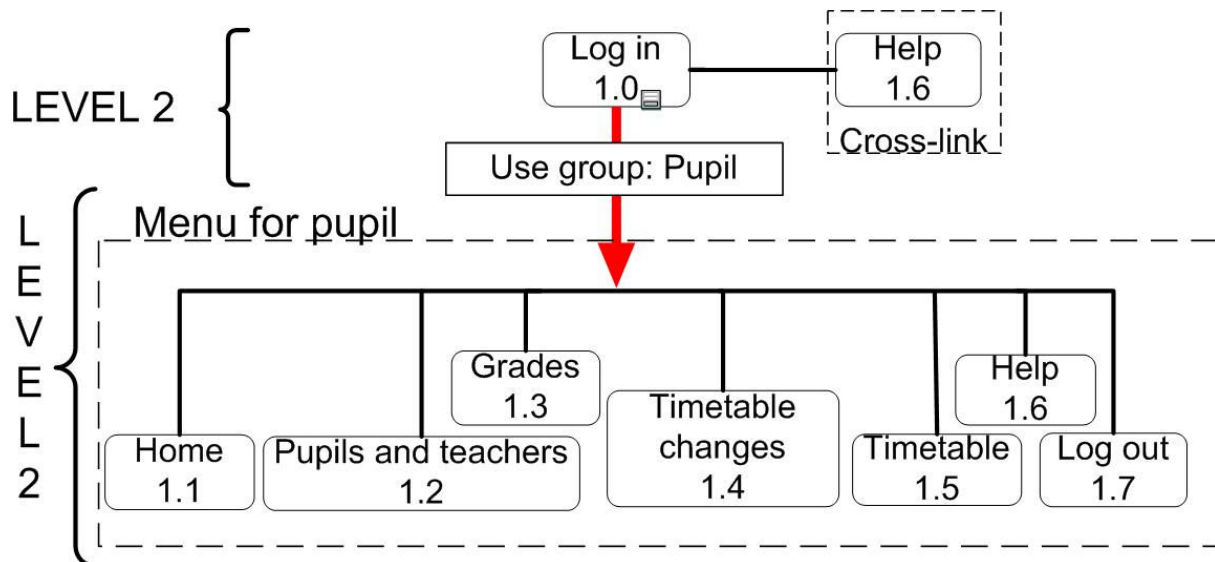


Figure 6.4.1-4 Simplified site diagram of pupil interface

#### 6.4.1.1 Feedback

- Polish version of the system required;
- Detailed information about content of each page required;
- Link allowing to go back the page required;
- Average for each kind in the first level grade view.

#### 6.4.2 Prototype two

##### Bilingualism

Implementation of prototype two was started with the changes due to language requirements. The global variable to hold the information about language between pages is session variable `$_SESSION['language']`. It is created when logging into the system and cannot be changed before logging out.

The variable is checked each time a text is to be displayed. There are three kinds of texts in the system:

- Information and instructions stored in the file "text.php",
- Information embedded in the code, ex. Names of the columns in tables,
- Information stored in database,

Bilingualism in each kind of texts had to be approached differently to prepare the system for bilingualism. Ex. the names of subjects, subject kinds, etc stored in database which need to be in two languages are recognized by the index to make the processing of language independent.

##### Class/pupil choice

It was decided that parents and teachers would have a choice of the information they want to browse. It means that if a parent has more than one child who attend the school, he/she will be able to choose the child whose grades and other information



the parent wants to browse. The parent may as well choose to browse the information about all his/her children.

In the case of a teacher, the choice of information means choice of the classes and subjects the teacher teaches. Ex. if a teacher John Smith teaches the following subjects:

- Polish Language in class 4a
- History in class 5a
- Polish Language in class 5a

The teacher can choose to see only information about the grades etc of class 5a from History. But the teacher can as well see all the information.

To implement this feature four session variable are used:

- **\$\_SESSION['children']** – stores logins, names and classes of all the children of a parent,
- **\$\_SESSION['chosen\_children']** – stores logins of the chosen children,
- **\$\_SESSION['classes']** – stores classes, subjects and index number of the record in class\_subjects table,
- **\$\_SESSION['chosen\_classes']** – stores indexes in **\$\_SESSION['classes']** of the chosen classes.

The session variables which are not used are initiated to zero. Therefore functions displaying information can be made universal for all the users. Functions recognize the way the information should be displayed checking the content on the session variables.

### **Grade interface**

**Figure 6.4.2-1** depicts the implementation of grade presentation and management in teacher's interface. It was implemented according to the design.



Figure 6.4.2-1 Implementation of grade presentation and management

### **User information in the header**

Teacher interface provides an extended information in the header, except for user's login and user's name, there are also shown the classes chosen by the teacher to work with and the subjects taught by the teacher in this classes.

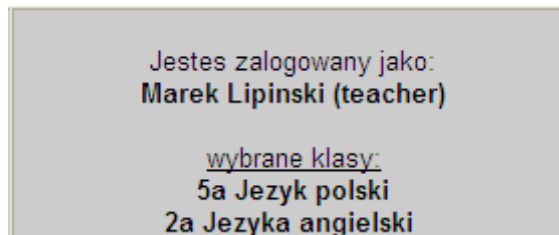


Figure 6.4.2-3 User info in teacher interface

Whereas in parent interface user information in the header shows the children chosen by the parent and the classes they attend.

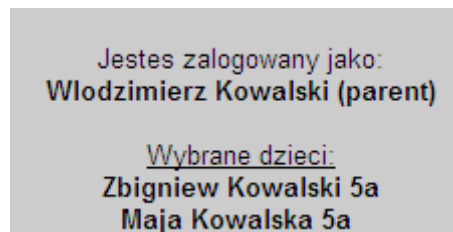


Figure 6.4.2-3 User info in parent interface

### **Instruction texts**

During this phase all the instructions and information texts for pupil, teacher and parent interface were consulted with the client and corrected by him.

### **HELP**

In this phase the help webpage was created. The page is available for all the users without the need to log in.

A simplified site diagram of teacher and parent interfaces are very similar to the pupils simplified sitemap presented in the previous chapter. A detailed site diagram of teacher interface is presented in the **Appendix C:C-1**.

#### **6.4.2.1 Feedback**

- Descriptive grades to be added,
- Possibility for teachers to add the same kind of grade to many pupils,
- Different colours of failed and excellent grades,
- Information about class parties, teachers' meeting with parents, news about class and school life should be available – not in the scope of the project,
- Teacher's remark about pupil's progress at school (individually), possibility for the teachers to send messages to parents – not in the scope of the project.

### 6.4.3 *Prototype three*

#### **Database**

Implementation of this prototype was started with database implementation which is described in Chapter **6.3 Database implementation**.

#### **Administrator interface**

Implementing administrator interface focused mainly on creating forms to enter, edit or delete all the information stored in the database. This theoretically easy task was very time consuming as there is quite considerable number of data which can be inputted into the system, edited or deleted. All the files which were created to manage administrator interface have prefix “admin”. Example screen shoots of admin interface with description can be found in **Appendix C:C-3**.

#### **Data import**

The most difficult in implementing administrator functionalities was the data import. Although the client uses a package of applications to manage all the administration related tasks, each application stores different data and enables different options of data export. The data which was used for testing data import can be found on **CD:Data to import**.

#### **Retrieving pupils’ and parents’ data**

Pupils data can be exported from clients application called “**Rejest uczenia**” (Eng. “Pupils register”). The exported file is meant for moving data between “Rejestr uczenia” applications and has “.sou” extension, which seems to be internal for the company. However data in the file is in the XML format. Therefore the import of pupils’ and parents’ data was performed using PHP XML functions.

#### **Retrieving teachers data**

Teachers data is managed by the client with the help of application called “**Arkusz Optivum**”. The most convenient way to retrieve data from “**Arkusz Optivum**” is to export the data to Excel file. There are available on-line open source PHP libraries which enables to read Excel files format, and one of such libraries is used (**Appendix C:C-4**). However, the Excel file created by “**Arkusz Optivum**” contains considerable number of unnecessary and redundant information which causes the file to be over 4MB. The desired information can be 55KB. As reading data from 4MB file is troublesome, it was decided that the client would need to select the appropriate information (detailed information provided to the user in the help) and copy it to a new file which can be imported to the system.

Such solution can be useful if the clients software changes export data structure. In such situation, it is just a matter of arranging the data in an appropriate way in the Excel file before uploading it to the system.

#### **Retrieval of timetable**

The applications used by the client to create timetable offers possibility to create a standalone website with the timetable. It is very convenient for the use in PSONline. However the timetable has to be imported twice. One of the requirements is to limit the access of pupils and parents to class timetables only. Thus it is important that pupils and parents have no access to the individual timetables of the teachers.

Client's software allows to export the entire timetable or the timetable of classes only. The export and import is done twice:

- Timetable of classes only
- Timetable of classes and teachers

Standalone timetable is composed of a catalogue tree with html files. It is not easy to upload catalogue tree, therefore the timetable needs to be zipped before export. POnline uploads zipped timetable, unzips the timetable, removes the old timetable, copies new timetable in the right location, performs necessary changes to incorporate the timetable into the system and reads through the html files to specify which html file represents which class and finally input the affiliation between files and classes to the database.

### **Data import problem**

During the implementation of data import a problem was met of Polish signs encoding. The imported data includes Polish character which were not being recognized during import process. Format conversions with standard functions were not successful. The solution was reached by encoding some of the Polish characters by standard functions and the others "by hand".

### **Automatic generation of login and password**

PSonline automatically generates logins and passwords for the imported users. Passwords are shown on the webpage directly after the data is imported. They need to be printed as the information is lost as soon as any other page is open.

#### **6.4.3.1 Feedback**

- Behaviour grades need to be added
- The right sequence of data import needs to be enforced
- The HELP access need to be restricted, namely, the parts concerning teacher interface and administrator interface should be available only for teachers and administrators.
- Easy mechanism of database clearing should be implemented
- Graphic grades needs to be added in levels 1-3

### **6.5 Final product**

The system is meant to be used and hold data of the school performance throughout one school year. Therefore the administrator is provided with a tool to reset the database to initial, empty state to prepare it for the import of new data. The password confirmation was implemented before the data is deleted from the database to prevent accidental deletions

The graphic grades were implemented and the grade interface changed appropriately.

All the changes indicated in the user feedback were made.

Final product implementation involved correcting mistakes in the texts, information and instructions in the system. Thanks to the method described in Chapter **6.2 File organization**, the client was able to correct most of the mistakes by his own.

## 7. Test

### 7.1 Test conduction

The tests were held throughout the prototyping process before a prototype was presented to the client. However, these tests were not formalised and not documented. It was agreed with the client that the formal tests would be conducted on the final version of the system. The test scenarios were agreed with the client after the third prototype was released. Sample basic scenarios are depicted below, the rest of this set of tests is enclosed in the **Appendix E:E-1** and other scenarios agreed with the client can be found in the **CD:Tests**

#### 1. Choose classes to browse information about

User input	System response
B1. User chooses "Wybierz klasy ktore chcesz przegladac" (Eng. "Choose classes to browse") from the menu	B2. System displays all the classes in which the teacher has lectures and what lectures he/she teaches with check box next to each choice class
B3. User ticks the check box next to appropriate class	
B4. User press the button "Wybierz" (Eng. "Choose")	B4. System displays information that the operation has been successful
	B5. In the "user information" part of the header (top, right corner) the chosen class with subjects are displayed

#### 2 Browse information about the chosen class (assuming that test 1. has been completed successfully)

User input	System response
B1. User chooses "Uczniowie i nauczyciele" (Eng. "Pupils and teachers") from the menu	B2. System shows all the information about chosen class, list all the pupils attending to the class and teachers teaching in the class
B3. User chooses to see information about a particular teacher by clicking teacher's name	B4. System displays detailed information about the chosen teacher
B5. User chooses to return to the previous window – he/she clicks "Wroc" (Eng. "Return")	B6. system returns to the previously displayed view
B7. User chooses to see detailed information about a particular pupil by clicking pupil's name	B8. System displays detailed information about the pupil, including information about pupils parents, if they are registered in the system

**3 Add grade (assuming that test 1. has been completed successfully)**

User input	System response
B1. User chooses "Pokaz, modyfikuj i dodawaj oceny" (Eng. "Show, modify and add grades")	B2. System displays all the grades of all the pupils in the chosen class for the current term
B3. User chooses "Dodaj ocene" (Eng. "Add grade") from the option column in the row where the name of the pupil he/she wants to add the grade to is	B4. System displays form "Dodaj ocene" (Eng. "Add grade")
B5. User chooses the appropriate grade from a list with all possible grades	
B6. User enters the date in the text box	
B7. User chooses the grade kind from the list with all available kinds	
B8. User can enter the description	
B9. User submits the grade by clicking button "Dodaj nowa ocene" (Eng. "Add new grade")	B10. System displays the data which has been added

**4 Edit grade (assuming that test 1. has been completed successfully)**

User input	System response
B1. User chooses "Pokaz, modyfikuj i dodawaj oceny" (Eng. "Show, modify and add grades") from the menu	B2. System displays all the grades of all the pupils in the chosen class for the current term
B3. User chooses "Edytuj ocene" (Eng. "Edit grade") from the option column in the row where the name of the pupil he/she wants to add the grade to is	B4. System displays "Zestawienie ocen z przedmiotu" (Eng. "Grades from the subject") screen
B5. User chooses the grade he/she wants to edit by clicking on the grade	B6. System displays "Edytuj istniejace oceny" ("Edit existing grades") screen, where all the fields are filled with the information about the chosen grade
B7. User chooses the grade from the list of available grades	
B8. User changes the date in the text box	
B9. User chooses the grade kind from the list with all available kinds	
B10. User changes the description	
B11. User submits the grade by clicking button "Zmien ocene" (Eng. "Change grade")	B12. System displays detailed data about the changed grade.

## 7.2 Test summary

The tests helped to detect considerable number of “small errors” in the program. The first test ratio was quite frightening with only 60% tests passed. However, it turned out that most of the mistakes were a consequence of constant project development and evolution. When subsequent changes were introduced to the system some areas of the system, affected by the changes, were not adapted appropriately. The mistakes were quite easy to correct and after the second iteration the ratio was almost satisfactory. Eventually a hundred percent ratio was reached and the PSonline\_Pack (**CD:PSonline\_Pack**) was sent for the tests by the client.

The client could not afford to perform all the tests, so he decided to choose randomly few and perform them. During the tests (by the client) on the real data a database problem was met. It seems that the number of data to check the integrity was too great and the database used by the client crashed while importing information about pupils and parents. It is not a mistake in the developed system but limited performance of the particular database. Tests on smaller number of data were positive. Tests with the InnoDB engine disabled were also positive. Therefore no changes were made in the system. It is recommended to check the performance of the system with different SQL database.

Another problem revealed by the client's tests was browser dependency.

The solution for the Polish signs encoding problems which works fine when tested by the developer in Internet Explorer, did not work correctly when the client performed tests in Mozilla Firefox. But changing to Internet Explorer solved the problem.

Eventually, with the InnoDB engine disabled and using Internet Explorer all the tests were completed with the ratio of 100%,



## 8. Conclusions

### 8.1 Achievements

All the project primary objectives stated at the beginning of the project were achieved, all primary and most of secondary requirements were met. However all of the requirements stated as optional and some low priority requirements were not fulfilled due to the time constraints only.

### 8.2 Future Work

The system is fully functional but is not ready for the usage by the client. As stated in the project scope, the Secure Socket-Layer (SSL), or similar encryption protocol necessary for such a system to go live, was not in the scope of the project. Therefore in the future it needs to be implemented. All the optional requirements, which were defined, are not difficult to add to the system but rather time consuming. It would be worth adding them to make the system even more attractive and functional. The following feature could be implemented:

- Communication between teachers and parents,
- Possibility for the administration to send messages to a desired user group
- Interface to add information about school events ,
- Problems with Polish letters encoding in Mozilla Firefox

In my opinion the interface for pupil could be more attractive and friendly looking.

### 8.3 Problems encountered

Problem	Solution
Polish letters encoding	problem solved for Internet Explorer, solution not provided for Mozilla Firefox
Reading Excel files	library supporting reading from Excel files was found
Uploading catalogue tree to the server	catalogue tree is uploaded compressed as zip file, decompressed at server and copied to the right location
Database hangs when too much data is imported	Turn off InnoDB engine

## 8.4 Project Evaluation

The project can be called successful due to the following:

- meeting most important client's requirements and objectives ,
- learning new skills and gaining experience in new fields.

It has to be clearly stated that this project was my first experience with web applications, PHP and HTML. Before starting my Final Year Project I had no experience in web-related programming. Having gained experience throughout development of the project I am aware now that many solutions used in the project might have been tackled differently. However, having no experience it was difficult to evaluate which solution is better for a given problem. It is possible that in many cases the lack of experience could have been substituted with through literature review and background reading. However the right balance between spending time studying theory and practising is always difficult to achieve.

A fine example of a wrong decision due to the lack of experience is the choice IDE. The entire project was developed in Notepad++ without any support of Integrated Development Environment neither for PHP script writing nor HTML interface creation. It would probably have saved a lot of time and effort to use even the simplest IDE. It would also allow to easily produce the project technical documentation. It also seems not entirely right decision to choose structural programming in stead of object orientated programming. Polymorphism and inheritance would probably prove to be very useful.

Thought the project development many obstacles were met. The most challenging was the import of data from the software used by the client which was one of the primary objectives. I am very happy that not only all the objectives concerning data import were met, but even more than planned was achieved. Except importing the data of all the users and information about the timetable, the information about affiliation between teachers-subjects-classes can be imported.

Although the project management appeared to be efficient and effective throughout the project the amount of work needed to complete the project on time fulfilling all the requirements was underestimated. Therefore not all the intended outcomes were achieved, ex. the English version was not finished. However, thanks to prioritizing of requirements, all of the most important requirements were fulfilled and the client seems to be satisfied (**Appendix F**).

The only deliverables stated in the project brief that was not achieved due to the time constraints and wrong IDE choice is the fact that technical documentation was not produced.

## List of References

- Agile Modelling (2006) *UML 2 Use Case Diagrams* [online] Available from <<http://www.agilemodeling.com/artifacts/useCaseDiagram.htm>>
- Booch, G.(2001) 'The architecture of Web applications'. *IBM Journal* [online]. Available from <[http://www.ibm.com/developerworks/ibm/library/it-boochoch\\_web/](http://www.ibm.com/developerworks/ibm/library/it-boochoch_web/)> [21 April 2008 ]
- Computerworld(2002) *System development live cycle* [online] Available online <<http://www.computerworld.com/action/article.do?command=viewArticleBasic&articleId=71151&pageNumber=1>>
- Corp21 (2005) *Web application technologies* [online] Available from <<http://www.corp21.com/WebApplications.pdf>> [18 April 2008]
- Cotterell, M. & Hughes, B. (1995) *Software Project Management*  
London: International Thomson Publishing Inc.
- Körös-Mikis, M.(2002) 'Defining Innovative Pedagogical Practice'. *English language studies and articles from OKI's R&D centres*. Available online <<http://www.oki.hu/oldal.php?tipus=cikk&kod=english-art-koros-defining> > [21 April 2008]
- Litwin, P. (1990) *Fundamentals of Relational Database Design*  
[online] Available from <<http://www.deepraining.com/litwin/dbdesign/FundamentalsOfRelationalDatabaseDesign.aspx> >[19 April 2008]
- PJB Associates(1997) *Multimedia in Education: The transition from primary to secondary Schools* [online] Available online<<http://www.pjb.co.uk/mmeduc.htm>>
- Project management Institute (2000) *PMI PMBOK*  
Penn.:Project Management Institute
- Silberschatz, A. (2001) *Database Computer Concepts*  
The McGraw-Hill
- Sommerville, I.(2007) *Software engineering 8*  
Harlow: Pearson Education Limited, Addison-Wesley
- Welling, L. & Thomson, L.(2005) *PHP i MySQL Tworzenie stron WWW. Vademecum Profesjonalisty (3rd edition)*. Warsaw: Wydawnictwo Helion
- Wieggers, K.(1999) 'First Things First: Prioritizing Requirements.'" *Software Development*, 7, (9)

## Bibliography

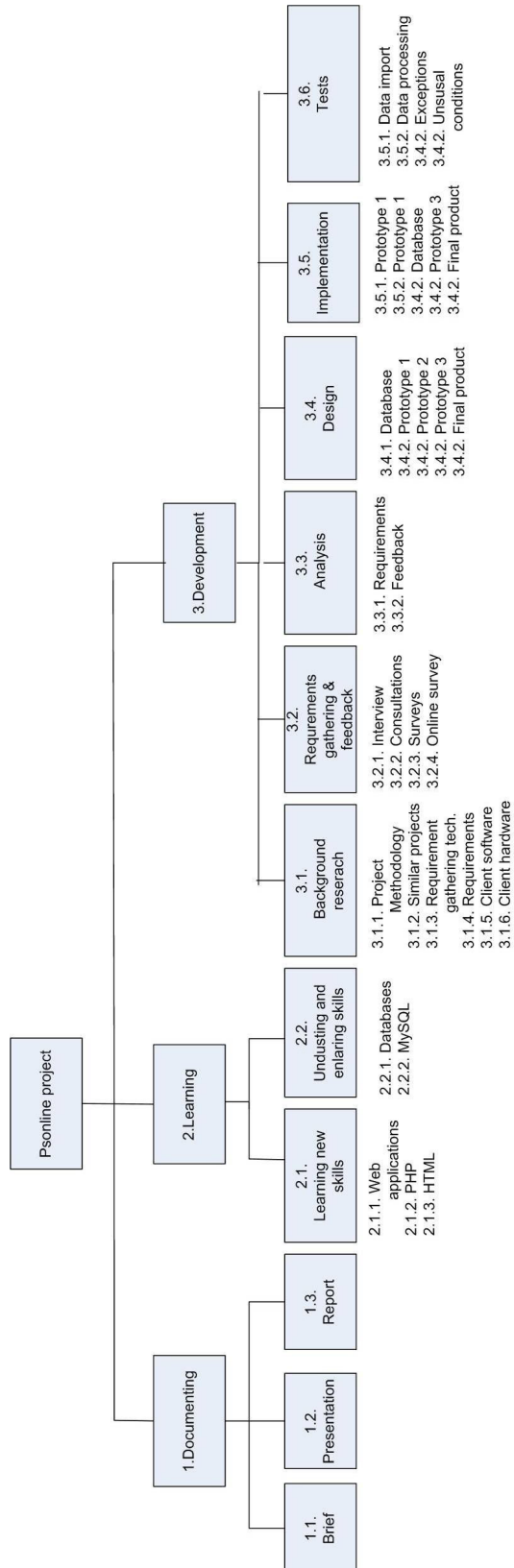
MySQL (2008) *MySQL 5.0 Reference Manual* [online] Available from  
< <http://dev.mysql.com/doc/refman/5.0/en/index.html> > [19 April 2008]

PHP(2008) *Documentation* [online] Available from  
<<http://www.php.net/docs.php>> [22 April 2008]

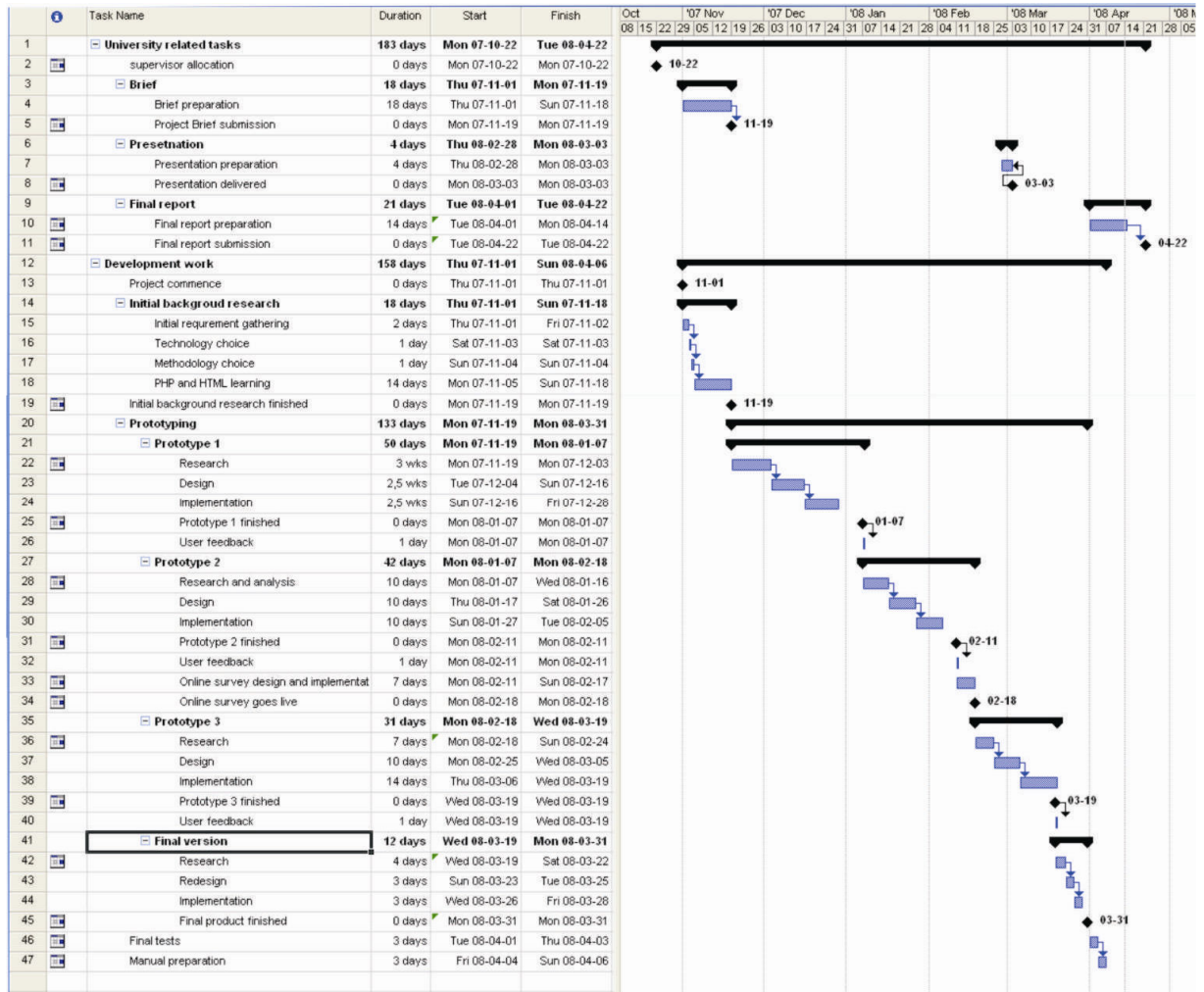
W3School (2008) *HTML Tutorial* [online] Available from  
< <http://www.w3schools.com/HTML/> >

# Appendix A – Project management

## A-1. WBS



## A-2. Project Schedule



**A-3. Project**

PSonline schedule of liaison with the client

Date	Milestone	Description	Task	People involved
1/11/2007		Collecting information about the clients idea of the system	Interview	Client
17/12/2007		Collection of requirements of intended users	Interview	Teacher Pupil Parent
19/11-3/12		Consultation and determining precise requirements	Consultations	Client
7/01/2008	Prototype 1 finished	Collecting feedback on prototype 1	consultation	Client
11/02/2008	Prototype 2 finished	Collecting feedback on prototype 2	consultation	Client
18/02/2008 for approx 2 weeks	Online survey goes live	Collecting feedback on prototype 2	Online survey	Teachers Pupils Parents
19/03/2008	Prototype 3 finished	Collecting feedback on prototype 3	Consultation, Test on real data	Client, teacher, pupil, parent
14/04/2008	Project approval			Client

# Appendix B – Liaison with the client and users

## B-1. Report form initial interview with the client

01/11/2007

## INTERVIEW 0

BASIC USER REQUIREMENTS GATHERING  
"MEETING" WITH THE CLIENT AND FUTURE ADMINISTRATOR (Skype)

**Questions:**

What should be the purpose of the system?  
Who should be the users?  
What are the functionalities the system should include?  
What are the system's desired features?  
What is the budget, is there any?  
What are the hardware school availabilities?

**Report after the interview:**

What should be the purpose of the system?  
The purpose of the system should be to simplify communication and information exchange between the school (teachers and administration) and its 'clients' (pupils and pupils' parents) as well as within the school (teachers – administration).

Who should be the users?

- pupils
- parents
- teachers
- administration staff


What are the functionalities the system should include?  
Show grades, timetable, timetable changes, messages sent by users

What are the system's desired features?  
Easy access from users homes, simple interface, easy usage, nothing special needed to user the system, the system should use the already existing data of the users on the administration computer.

What is the budget, is there any?  
No budget at all, the software should be free, the server should be made at school using the existing computer and internet connection.

What are the hardware school availabilities?  
A computer can be provided out of the school existing resources. The internet connection exists, the details of internet connection are enclosed.

**SZKOŁA PODSTAWOWA NR 11**  
im. Juliusza Słowackiego  
44-270 Rybnik 10, ul. Boczna  
tel. 42-29-950, fax 42-29-965  
Regon 000728078, NIP 642-27-07-251

  
**DYREKTOR SZKOŁY**  
mgr Maciej Lipiński



## B-2. Report from the second interview with the client

17/12/2008

# INTERVIEW 1

With the client and future administrator.  
General overview of the system and specific questions about pupil's interface

**Introduction**

*I conducted the research to find out the best technology and way to implement system described in the previous interview. In my opinion the best way is to create a web application. It will allow users free access from anywhere the Internet is available. No additional software on client's side is needed. The system can be developed and maintained with open source tools. The only cost in such a solution is the hardware. However, the server can be created out of one of the school's computers. However, in such case it would be good to change the internet connection kind.*

**Questions:**

*What are the functionalities the system should have for each particular user group?  
What should be the benefits for the users?  
Should pupils see the timetables of teachers, or should their view be limited to class timetable only, or their class timetable?  
What information about grades should be stored in the database?  
Should the grades be seen by the limited number of pupils, what should be the restriction on data access in the system?  
Is all the data available for import form the user software needed to be imported into the system?*

**Report after the interview:**

What are the functionalities the system should have for each particular user group?

- Pupils can see: classmates and teachers, grades, timetable and timetable changes
- Parents can see information about all their children attending the school, children's classmates and teachers, children's grades, children's timetable and timetable changes,
- Teachers can see information about the classes the teacher teach (class classmates and other teachers ), add, edit and delete children's grades, timetable and timetable changes,
- Admin can see, add, edit, delete information about all users, classes, timetable changes, can import data about users and timetable.

What should be the benefits for the users?

- Pupils should have easy access to the information. This system can be a good reason for them to learn how to use computer and internet.
- Parents should have an opportunity to monitor their children progress at school receiving the information first-hand from school without bothering to visit school. It should be an easy way for the parents to receive information from school but it should not be a substitute for parents meeting.
- Teachers should benefit by obtaining an easy mean of communication with pupils and parents, however, this intended user group will also find additional duties connected with the system.
- School administration – easy control of pupils and teachers, at any time the administrator can check any particular pupil's grades. Easy communication with pupils, teacher and-parents. The system will also impose additional duties on administration.

Should pupils see the timetables of teachers, or should their view be limited to class timetable only, or their class timetable?

Pupils should see their class timetable, however there should be a possibility to see other classes timetables as well. They should not see teachers' timetables

What information about grades should be stored in the database?

Grade, the date it was received by the pupil, who received it and who gave it, why it was given, the subject it was given from and the event (ex. test, writing).


Should the grades be seen by the limited number of pupils, what should be the restriction on data access to the system?

Each user should see data only connected with the user, ex. grades should be seen only for the child which received the grade and for the teacher who entered the grade.

Is all the data available for import form the user software needed to be imported into the system?

Yes

SZKOLA PODSTAWOWA NR 22  
im. Juliusza Słowackiego  
44-270 Rybnik 10, ul Boczna  
tel. 42-25-950, fax 42-29-965  
Regon 000728078, NIP 642-27-07-251

DYREKTOR SZKOŁY  
  
mgr Marcin Lipiński

### **B-3. Collective report form interviews with a rep. of teachers, parents and pupils**

17/12/2007

## INTERVIEW 2

With the rep. of teachers, parents and pupils  
General overview of the system and specific questions about pupil's interface

Introduction

The school wants a virtual online primary school administration system to be created. A system should simplify communication and information exchange between the school (teachers and administration) and its 'clients' (pupils and pupils' parents) as well as within the school (teachers – administration). Each user will have her/his own account with the interface and information relevant to the group affiliation (pupils, parents, teachers, administration).

Questions:

How do you like an idea of such a system?  
What are the functionalities the system should include in your opinion?  
What are the functionalities you would appreciate from the system?  
How could the system help you?  
What are the benefits you gain from the system, in your opinion?  
Would you need such a system?

Report after the interview:

How do you like an idea of such a system?  
All interviewees generally liked the idea, the teacher pointed out that it could mean more work for teachers.

What are the functionalities the system should include?  
Possibility to see timetable, browse grades by parents and pupils, compare grade with the others without seeing the other grades (parents), see contact details of children and parents (by teachers), see information about school events, create online newspaper, relations from school events.

What are the functionalities you would appreciate from the system?

The functionality of the system useful for a teacher:

- Calculation of pupils average of grades
- Possibility to view timetable
- Possibility to view timetable changes

The functionality of the system useful for a pupil:

- Possibility to see all grades and compare grade with the pupils
- Possibility to view timetable and timetable changes
- Easy to use.
- Possibility to see information about school events

The functionality of the system useful for a parent:

- Possibility to compare his/her child's grades with the rest of the class
- Possibility to view timetable and timetable changes at any time and anyway, thus having a greater control over their child
- Possibility to see information about each grade and what it is for, why was it given

How could the system help you?

Benefits from the system the teacher has pointed out:

- Calculate the grade averages which has to be done by hand now
- Access to information from home, ex. timetable and its changes,
- Possibility to compare the individual pupil grades with the entire class, useful during parent meetings
- Print the grades of pupils to hand in to the parents at parents meetings

Benefits pointed out by the pupil:

- Easy control of grades
- "My friends at other schools will be envy..."

The functionality of the system useful for a parents:

- control of child performance at school without the trouble of visiting school
- can easily verify the information their children give them.

Za Radę Rodziców:                      Rada Pedagogiczna:                      Samorząd Uczniowski  
 Krzysztof Wymań                      Janusz Kopycki                      Alex Dziadek

### B-4 Example how the timetable changes are displayed by the client currently

DATA :		04.12.2006 r. - poniedziałek												
<b>ZASTĘPSTWA ZA NIEOBECNYCH NAUCZYCIELI</b>														
Lp.	Nazwisko i imię nauczyciela, przedmiot	Symbol naucz.	Powód nieob.	LEKCJA - KLASA							ZASTĘPSTWO :			
				Platne / Nieplatne / Zwolnić / Odmówić							Dyżur pełni	Czas trwania	Podpis	
				1	2	3	4	5	6	7				
1.	teacher kształcenie zinteg.	SE	opieka nad dz. do l.14	1b odm.	1b odm.	1b RJ p	1b NJ p					<b>PJ</b>		
2.	teacher kształcenie zinteg.	RA	udział w komisji na nauczyciela dyplom.					3a zwol.				<b>HB</b>		

DATA :		12.12.2006 r. - wtorek													
<b>ZASTĘPSTWA ZA NIEOBECNYCH NAUCZYCIELI</b>															
Lp.	Nazwisko i imię nauczyciela, przedmiot	Symbol naucz.	Powód nieob.	LEKCJA - KLASA								ZASTĘPSTWO :			
				Platne / Nieplatne / Zwolnić / Odmówić								Dyżur pełni	Czas trwania	Podpis	
				1	2	3	4	5	6	7	8				
1.	teacher kształcenie zinteg.	RJ	opieka nad dzieckiem do l.14					2a zwol.					<b>GJ</b> parter	7.45 - 8.00	
2.	teacher muzyka/rytmika	ŚM	konkurs "Karolinka"	6c odm.	5c CB n	4b LG n		0 Mb p					<b>FZ</b> parter	13.30 - 13.40	
3.	teacher informatyka/technika	DJ	L - 4 (opieka)	5a LM p	6c/1 LM p	6c/2 LM p	5c SE p	5a SE p	5b SE p				<b>LA</b> I piętro	7.45 - 8.00	
													<b>ZI</b> I piętro	8.45 - 8.55	
													<b>LG</b> I piętro	9.40 - 9.50	
													<b>PJ</b> I piętro	10.35 - 10.45	
													<b>GJ</b> I piętro	11.30 - 11.45	



**B-5. Table with prioritized requirements**

	Requirements
<b>H I G H</b>	<p>The system <b>shall</b> have 4 user groups: administrator, pupils, teachers, parents.</p> <p>The system <b>shall</b> allow users to see school related information concerning them</p> <p>The system <b>shall</b> be easy to use and easily available and accessed from users homes.</p> <p>Any information in the system shall be accessed only after logging in.</p> <p>The system <b>shall</b> allow easy import of required data from the software used by the client,</p> <ul style="list-style-type: none"> <li>• Login structure and generation method</li> <li>• Teachers' access only to the grades the pupils have received in the subject they teach</li> <li>• Access to "help" possible without log in</li> <li>• Data to be stored by the system exactly determined</li> <li>• Automatic generation of passwords for imported users</li> <li>• Administrator's possibility to edit and delete all data which can be entered or imported</li> </ul>
<b>M E D I U M</b>	<p>Each user group shall have different user interface, authority and system options available</p> <ul style="list-style-type: none"> <li>• Grades grouped in categories</li> <li>• Grade categories extensible</li> <li>• Calculation of individual and class grade average</li> <li>• Teacher can determine the class to work with</li> <li>• Parents can determine the child/children to work with</li> <li>• Administration's possibility to add/edit/delete grades.</li> <li>• Access by teachers to all detailed user data</li> <li>• Possibility to add grade to multiple pupils simultaneously</li> </ul>
<b>L O W</b>	<ul style="list-style-type: none"> <li>• Each user login, name and surname displayed in header</li> <li>• Teachers' and administrator's possibility to send messages to parents</li> <li>• no requirement that on opening the timetable website</li> <li>• System interface bilingual</li> <li>• upon logging in to the system</li> </ul>
<b>O P T I O N A L</b>	<p>Information exchange between users</p> <p>General information and messages from administration to particular user group</p> <p>Online school event chronicle</p> <p>Online newspaper with editing tools</p>

### B-6 Screen shot of the online questionnaire with description

**Ankieta - PSONline**  
**Drogi nauczycielu!**

Chcemy usprawnić komunikację między Szkołą (Wami) a Rodzicami i Uczniami.

W tym celu pokusiłmy się o uruchomienie tej strony internetowej. W dobie powszechnej komputeryzacji taka forma przekazywania informacji staje się konieczna, bo jest szybka i skierowana do konkretnego odbiorcy. Prosimy, wypróbujcie i sprawdźcie możliwości naszej Wirtualnej Szkoły. Podzielcie się z nami Waszymi wrażeniami. Żeby Wam ułatwić zadanie przygotowaliśmy poradnika ankietę. Ta ankieta ma na celu sprawdzić, czy nasza Wirtualna Szkoła spełnia Wasze oczekiwania. Chcemy się dowiedzieć: Czy nasza strona jest prosta i łatwa w obsłudze? Co na naszej stronie podoba Wam się? Jaki informacje są dla Was ważne i potrzebne? Jakie jeszcze inne informacje (Waszym zdaniem) powinny się znaleźć na stronie dostępnej dla Nauczycieli, czyli Waszej stronie?

Wasze opinie i uwagi będą uwzględnione przy konstruowaniu nowszej wersji Naszej Wirtualnej Szkoły. Poproście swoich Uczniów i ich Rodziców, żeby też wypełnili ankietę umieszczoną na ich stronach Wirtualnej Szkoły. Ich zdanie też jest dla nas ważne!

Dziękujemy za strasne wypełnienie ankiety

**Krotka instrukcja jak wypełniać ankietę:**  
 Ankieta dotyczy Systemu Wirtualnej Szkoły Postawowej PSONline. Aby wejść do systemu kliknij poniższy link, zostaniesz zalogowany do systemu w nowym oknie. W ankiecie znajdują się również odnośniki do różnych zakładek systemu, które zostaną otworzone w nowym oknie. Po uzupełnieniu ankiety kliknij przycisk "Wyślij", aby przesłać Twoje odpowiedzi na serwer. Po wylogowaniu lub zamknięciu strony, Twoje odpowiedzi nie będą zapamiętane, tak więc ankietę musisz wypełnić za jednym razem w całości.

Aby zalogować się do systemu PSONline, kliknij [tutaj](#)

Link to PSONline

**ANKIETA**

Short open question

1. Jakiego przedmiotu uczysz?

2. Proszę kliknąć [tutaj](#) aby otworzyć system „pobaw” się troszkę i powiedz jakie jest Twoje pierwsze wrażenie

3. Jeśli w poprzednim polu zaznaczyłeś/as "inna odpowiedź", w poniższym polu opisz swoje pierwsze wrażenie

Closed open question

4. Czy Twoim zdaniem system jest łatwy w obsłudze?

5. Jeśli w poprzednim polu zaznaczyłeś/as "nie", w poniższym polu opisz dlaczego tak sądzisz.

Open question

6. Czy możliwość wyboru klas, w których zmiany możesz wprowadzać i przeglądać informacje ([tutaj](#)) jest Twoim zdaniem przydatna?

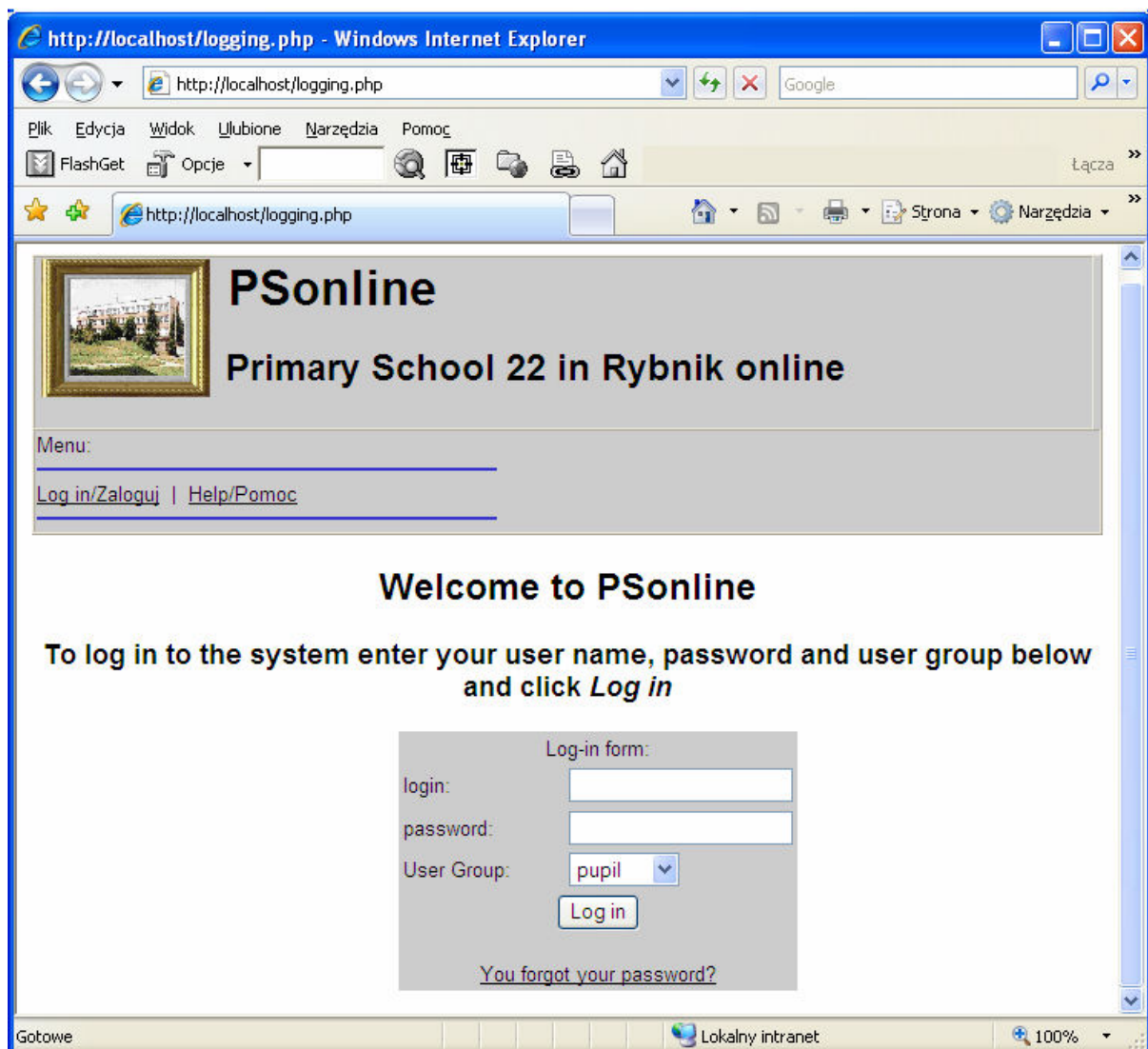
7. Jeśli w poprzednim polu zaznaczyłeś/as "inny pomysł", w poniższym polu podaj jakie jest dla Ciebie najlepsze rozwiązanie

8. Czy możliwość zobaczenia informacji o innych nauczycielach (ale również o Tobie przez innych nauczycieli, ([tutaj](#)) jest Twoim zdaniem przydatna?

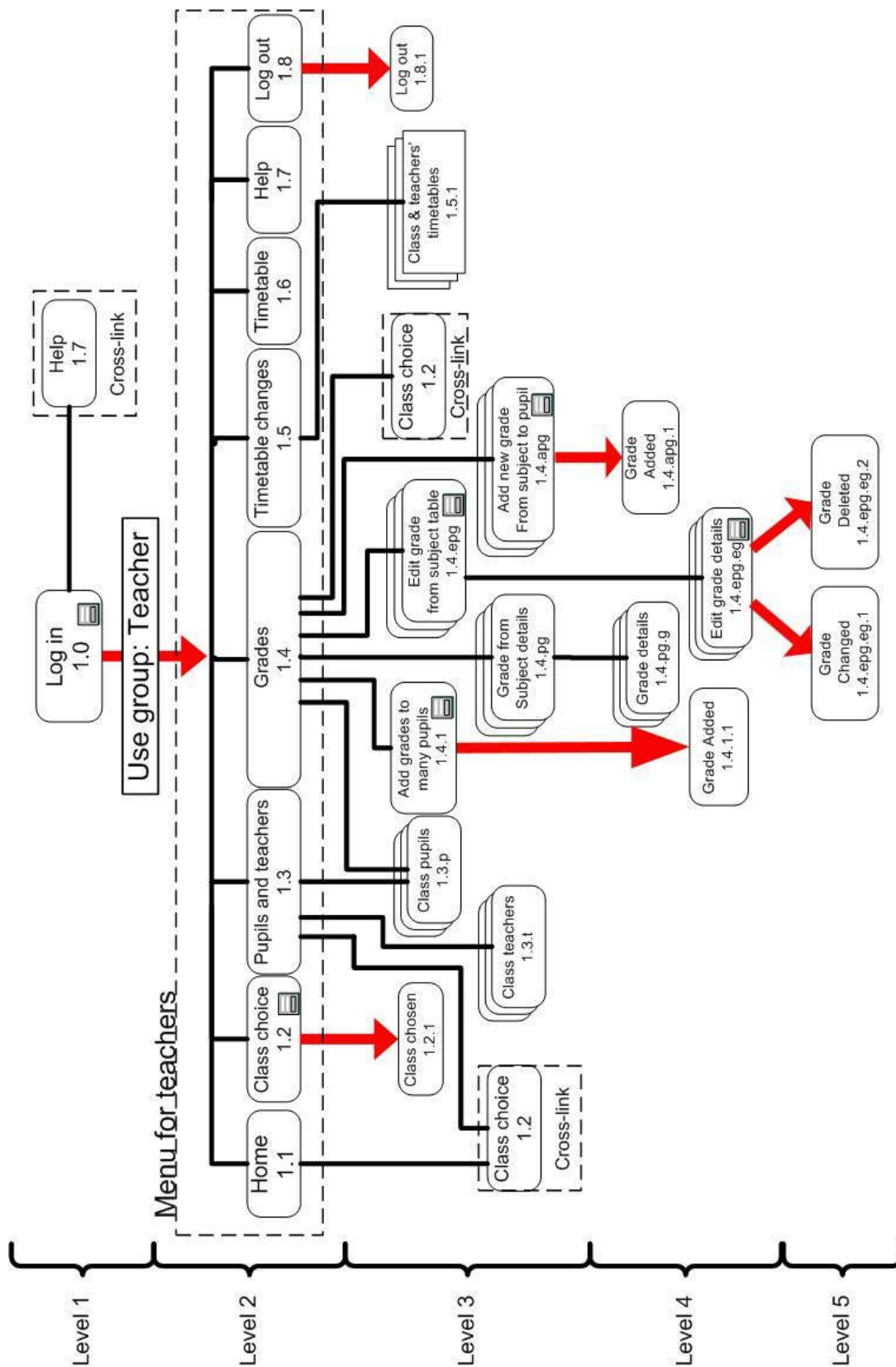
Instruction

## Appendix C – Instruction on POnline installation

### C-1: Login screen



**C-2: Web Diagram of teacher interface**



**C-3: Example screenshot of administrator interface with descriptions**

**PSONline**  
**Szkola Podstawowa nr 22 online - interface administratora**

Jestes zalogowany jako: maciej Lipinski (lipinsk3)

Wyloguj | Help/Pomoc

Zmien haslo | Import danych

Aktualne zmiany plany zajec | Plan zajec

Pokaz, modyfikuj i dodawaj oceny | Aktualne zmiany plany zajec

Przegladaj uzytkownikow | Wtorek

Wybierz klasy ktore chcesz przegladac | Poniedzialek

Uczniowie i nauczyciele | Sroda

Wybierz dzien: | Czwartek

Strona glowna | Piątek

Wybierz przegladac | Sobota

**Aktualne zmiany: 20/04/2008 - Poniedzialek**

Usun wszystkie zmiany z tego dnia

Klasa	LEKCJE									
1a	-	-	-	-	-	-	7	-	8	9
							Krzyzstof Sass			

**Zastepstwa za nieobecnosci nauczycieli**

Lp.	Nazwisko i imie nauczyciela, przedmiot	Symbol nauczyciela	Powod nieobecnosci	LEKCJE - KLASA												
				1	2	3	4	5	6	7	8	9				
1	Marek Lipinski Jezyk polski, Jezyka angielski, Edukacja jezykowa, Zachowanie.	ML	costam									1a Krzyzstof Sass paid			usun	
2	M Lipinski											M Lipinski			platny	Dodaj

**Zastepstwo**

Dywan pelni	Czas trwania
Marek Lipinski parter	14:00

Annotations:

- Timetable changes**: All users can see this view
- Timetable changes**: Table seen only by teachers and admin – changes in this table are reflected in the above table
- Teachers' duty substitution**: Table viewed only by teachers and admin
- Absent teacher**: Points to the 'Powod nieobecnosci' field.
- Absence reason**: Points to the 'Symbol nauczyciela' field.
- In which class**: Points to the 'Symbol nauczyciela' field.
- Place of substitution**: Points to the 'Dywan pelni' field.
- Kind of substitution**: Points to the 'Czas trwania' field.
- Substituting teacher**: Points to the 'M Lipinski' dropdown in the 'LEKCJE - KLASA' table.
- Time of substitution**: Points to the '14:00' field in the 'Zastepstwo' table.



## C-4: Information about PHP-ExcelReader

ExcelReader – library used in the system to read Excel files. The library can be found in **CD:Implementation/Excel**.

```
2  /* vim: set expandtab tabstop=4 shiftwidth=4 softtabstop=4: */
3
4  /**
5   * A class for reading Microsoft Excel Spreadsheets.
6   *
7   * Originally developed by Vadim Tkachenko under the name PHPExcelReader.
8   * (http://sourceforge.net/projects/phpexcelreader)
9   * Based on the Java version by Andy Khan (http://www.andykhan.com). Now
10  * maintained by David Sanders. Reads only Biff 7 and Biff 8 formats.
11  *
12  * PHP versions 4 and 5
13  *
14  * LICENSE: This source file is subject to version 3.0 of the PHP license
15  * that is available through the world-wide-web at the following URI:
16  * http://www.php.net/license/3\_0.txt. If you did not receive a copy of
17  * the PHP License and are unable to obtain it through the web, please
18  * send a note to license@php.net so we can mail you a copy immediately.
19  *
20  * @category Spreadsheet
21  * @package Spreadsheet_Excel_Reader
22  * @author Vadim Tkachenko <vt@apachephp.com>
23  * @license http://www.php.net/license/3\_0.txt PHP License 3.0
24  * @version CVS: $Id: reader.php 19 2007-03-13 12:42:41Z shangxiao $
25  * @link http://pear.php.net/package/Spreadsheet\_Excel\_Reader
26  * @see OLE, Spreadsheet_Excel_Writer
27  */
28
```

## Appendix D –PSonline installation instruction

*PS*ONLINE

# Primary School Online

Maciej Lipiński

Instrukcja instalacji systemu *PSonline*

Wszystko, co jest potrzebne do uruchomienie i działania **PSonline**, znajduje się w **PSonline\_Pack**. Do uruchomienia systemu potrzebne będą następujące pliki/katalogi:

- **WAMP5** – darmowy serwer **Apache** i baza danych **MySQL**
- **PSonline** – katalog zawiera zbiór skryptów PHP stanowiący system **PSonline** i dane potrzebne do jego działania, konfiguracji i uruchomienia bazy danych

1. Instalacja pakietu **WAMP5**:

- a. **WAMP5** 1.7.3 znajduje się w dostarczonym pakiecie,
- b. Przed instalacją programu należy utworzyć w dobrze dostępnym miejscu (np. C:\) katalog o nazwie „**PSonline**”. W tym katalogu należy utworzyć podkatalog: **DocumentRoot**.
- c. Należy zainstalować **Serwer**.
- d. Podczas instalacji jako „**WWW directory**” należy ustawić katalog **PSonline/DocumentRoot**
- e. Należy również ustawić jako przeglądarkę domyślną **Internet Explorer**
- f. Szczegółowych informacji na temat pakietu **WAMP5** i jego instalacji należy szukać na stronie: <http://www.wampserver.com/en/>

2. Tworzenie odpowiedniej struktury katalogów i kopiowanie systemu plików:

- a. Jeśli wskazówki z punktu 1. zostały prawidłowo wykonane, po otwarciu „**WWW directory**” tak jak na Rys.1 powinien odtworzyć się katalog o następującej ścieżce:

[zależna od użytkownika]\PSonline\DocumentRoot\



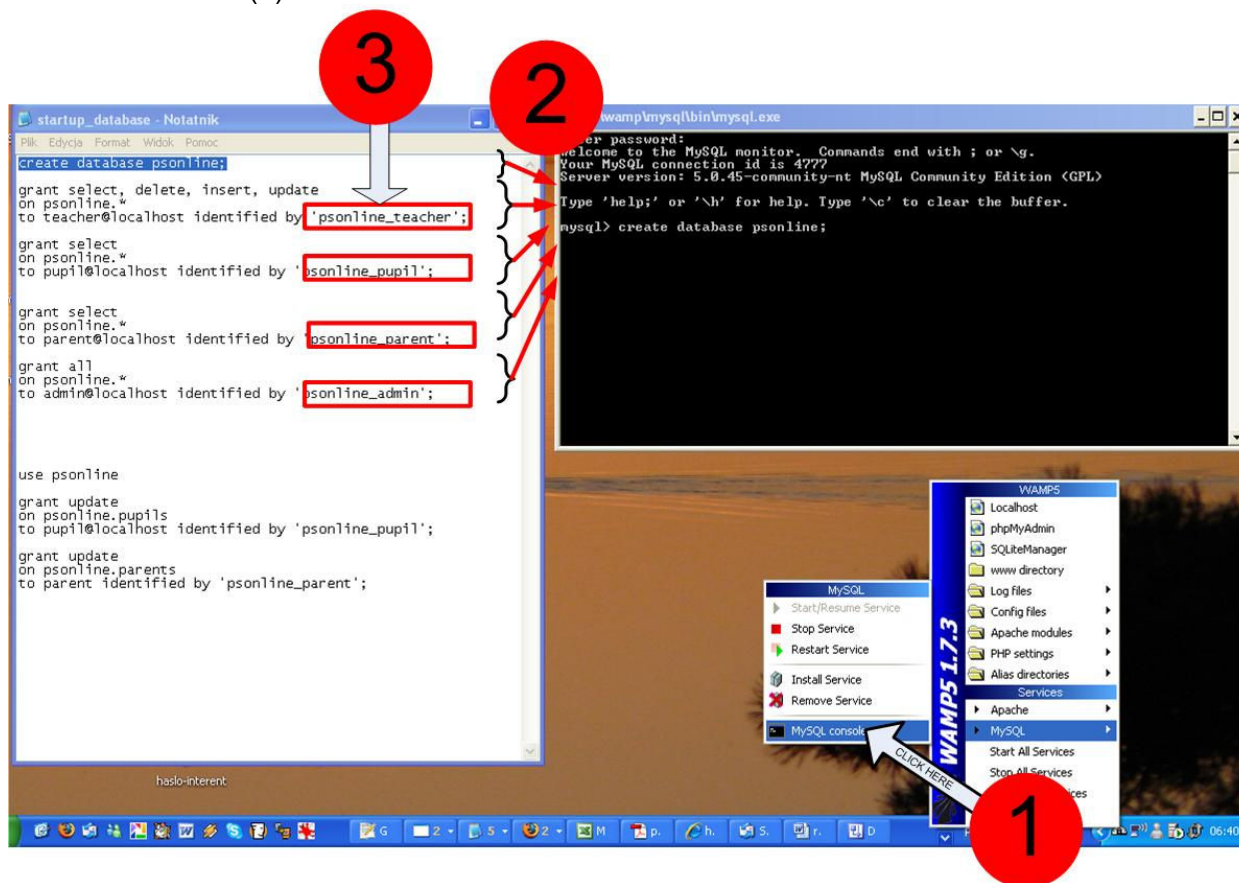
Rysunek 1

- b. Po upewnieniu się, że drzewo katalogowe ma odpowiednią strukturę, należy zamienić utworzony w punkcie 1. katalog **PSonline** na katalog **PSonline** (wraz z całą jego zawartością) znajdujący się w pakiecie **PSonline (PSonline\_Pack)**
- c. Aby sprawdzić czy dotychczasowe działania zostały wykonane prawidłowo, należy otworzyć **Localhost** klikając na odnośnik „**Localhost**” w menu **WAMP5**. Jeśli w przeglądarce zostanie otworzona strona logowania programu **PSonline**, to znaczy, że wszystko na razie zostało dobrze wykonane.

3. Zanim nie zostanie skonfigurowana baza danych, **PSonline** nie będzie działać - baza danych jest jego niezbędną częścią. Utworzenie i konfiguracja bazy danych:

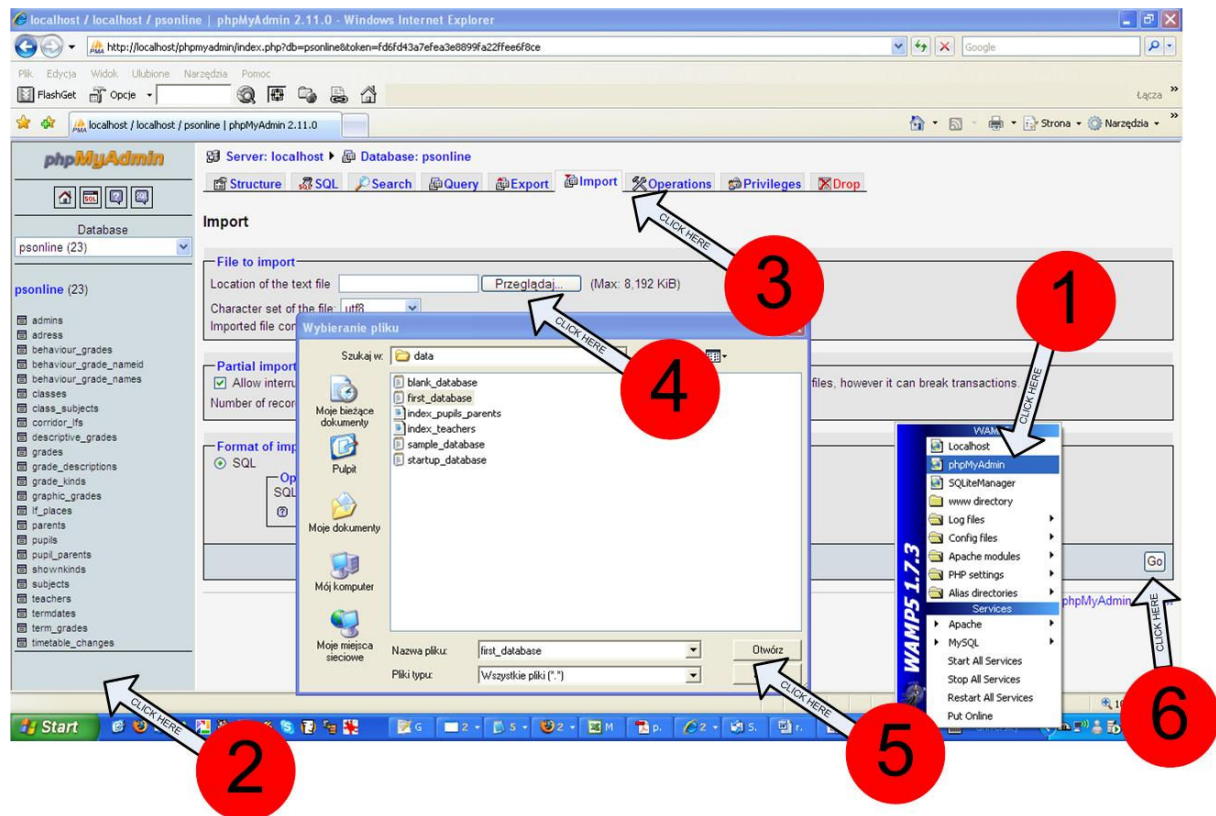
- a. Do konfiguracji i utworzenia bazy danych niezbędne są dwa pliki znajdujące się w katalogu **'data'** w folderze **'PSonline'** w pakiecie: **'startup\_database.sql'** i **'first\_database'**.
- b. Należy otworzyć plik **'startup\_database.sql'** i wykonać następujące czynności pokazane na Rys.2:
  - i. Otworzyć konsolę **MySQL** (1)

- ii. Zalogować się do utworzonej podczas instalacji bazy danych **MySQL**
- iii. Skopiować, tak jak to jest zilustrowane, kolejne polecenia **MySQL** do konsoli, następnie nacisnąć **'ENTER'** w celu ich wykonania. Dotyczy to pierwszych 5 poleceń (2)
- iv. **UWAGA:** zaznaczone na czerwono fragmenty kodu, to hasła dla każdego z użytkowników sprawdzane podczas ich dostępu do bazy danych. Jeśli program jest instalowany jedynie w celach próbnych, nie należy zmieniać tych haseł, jednak przy zastosowaniu z prawdziwymi danymi konieczna jest zmiana haseł na bardziej „nieoczywiste”. Te same hasła będą musiały być wprowadzone w samym systemie **POnline** w pliku konfiguracyjnym (3).



Rysunek 2

- c. Następnie należy otworzyć **phpMyAdmin** w przeglądarce **WWW** i importować szkielet bazy danych z podstawowymi danymi w następujący sposób (Rys.3):
  - i. Z menu **WAMP5** otworzyć odnośnik **'phpMyAdmin'** (1)
  - ii. Wybrać utworzoną przed chwilą bazę danych **'psonline'** (2)
  - iii. Otworzyć zakładkę **'import'** (3)
  - iv. Otworzyć plik **'first\_database.sql'**(4 -> 5)
  - v. przycisnąć **'wykonaj'/'go'**(6)
- d. Przekopiuj trzy pozostałe komendy z Rysunku 2 do konsoli



Rysunek 3

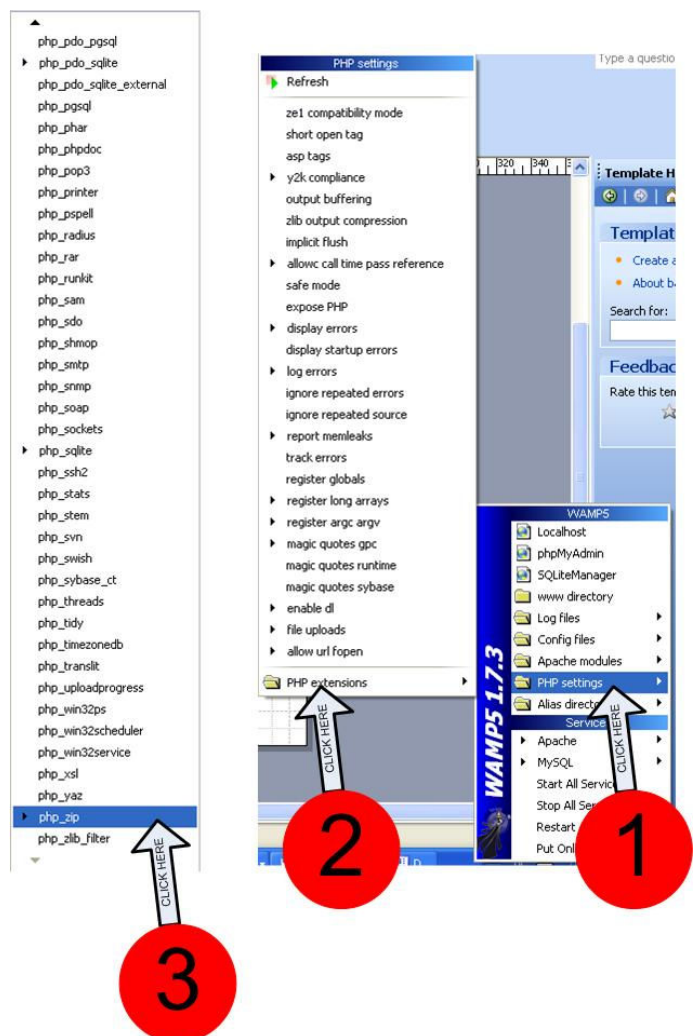
4. Konfiguracja serwera PHP -

Aby **POnline** prawidłowo działał potrzebne jest uruchomienie obsługi formatu 'zip', aby to zrobić należy wybrać z menu **WAMP5** zakładkę '**PHP settings**'(1), następnie wybrać kolejną zakładkę '**PHP extentions**' i z rozwijanej listy zaznaczyć jeden z ostatnich możliwych parametrów o nazwie '**php\_zip**'

5. Gratuluję, baza danych została utworzona z przykładowymi użytkownikami:

- a. **Admin**: login '**lipinsk3**', hasło: '**lipinsk3**'
- b. **Uczeń** : login '**pupil**', hasło: '**pupil**'
- c. **Rodzic** : login '**parent**', hasło: '**parent**'
- d. **Nauczyciel**: login '**teacher**', hasło: '**teacher**'

6. Aby się zalogować, należy otworzyć z menu **WAMP5** odnośnik '**Localhsot**', który otworzy stronę logowania.



## Appendix E – test

Basic scenarios of tests of user interface

### 5 Add multiple grades (assuming that test 1. has been completed successfully)

User input	System response
B1. User chooses “Pokaz, modyfikuj i dodawaj oceny” (Eng. “Show, modify and add grades”)	B2. System displays all the grades of all the pupils in the chosen class for the current term
B3. User chooses “Dodaj oceny” (Eng. “Add grades”) from the option column	B4. System displays form “Dodaj nowe oceny” (Eng. “Add new grades”) screen
B5. User enters from the lists of available grades next to the pupils’ names the appropriate grade or makes choice	
B6. User enters date in the text box	
B7. User chooses grade kind from the list of all available kinds	
B8. User can enter the description	
B9. User submits the grades by clicking button “Dodaj nowe oceny wielu uczniom” (Eng. “Add new grades to multiple pupils” )	B10. System displays names of the pupils who received the grades, the grades received and the details about the grades

### 6 Add term grades (assuming that test 1. has been completed successfully)

User input	System response
B1. User chooses “Pokaz, modyfikuj i dodawaj oceny” (Eng. “Show, modify and add grades”)	B2. System displays all the grades of all the pupils in the chosen class for the current term
B3. User chooses “Dodaj” (Eng. “Add”) from the “Ocena semestralna” (Eng. “Term grade”) column in the row where the name of the pupil he/she wants to add the grade to is	B4. System displays form “Dodaj nowa ocene” (Eng. “Add new grade”)
B5. User enters from the lists of available grades the appropriate grade	
B6. User enters date in the text box	
B7. User submits the grades by clicking button “Dodaj nowa ocene” (Eng. “Add new grade”)	B8. System shows all the grades of all the pupils in the chosen class for the current term with the new grade added

**7 Delete grade (assuming that test 1. has been completed successfully)**

User input	System response
B1. User chooses "Pokaz, modyfikuj i dodawaj oceny" (Eng. "Show, modify and add grades")	B2. System displays all the grades of all the pupils in the chosen class for the current term
B3. User chooses "Edytuj ocene" (Eng. "Edit grade") from the option column in the row where the name of the pupil he/she wants to add the grade to is	B4. System displays "Zestawienie ocen z przedmiotu" (Eng. "Grades from the subject") screen
B5. User chooses the grade he/she wants to delete by clicking on the grade	B6. System displays form "Edytuj istniejace oceny" ("Edit existing grades"), where all the fields are filled with the information about the chosen grade
B7. User click the button "Usun" (Eng. "Delete")	B8. System displays the message that the grade has been deleted successfully

**8 Delete term grades (assuming that test 1. has been completed successfully)**

User input	System response
B1. User chooses "Pokaz, modyfikuj i dodawaj oceny" (Eng. "Show, modify and add grades")	B2. System displays all the grades of all the pupils in the chosen class for the current term
B3. User chooses "Usun" (Eng. "Delete") from the "Ocean semestralna" (Eng. "Term grade") column in the row where the name of the pupil he/she wants to delete the grade from is	B4. System displays all the grades of all the pupils in the chosen class for the current term without the deleted grade

**9 Display timetable**

User input	System response
B1. User chooses "Plan zajec" (Eng. "Timetable") from the menu	B2. System opens new browser window with timetable

**10 Show timetable changes for Monday**


User input	System response
B1. User chooses "Aktualne zmiany planu zajec" (Eng. "Current timetable changes") from the menu	B2. System displays two tables with timetable changes for the current day
B3. User chooses "Poniedzialek" (Eng. "Monday" ) from the additional menu	B4. System displays two tables with timetable changes for Monday

**11 Change password**

User input	System response
B1. User chooses "Zmien haslo" (Eng. "Change password") from the menu	B2. System displays the change menu form
B3. User enters the old password into text box labelled "Stare haslo"(Eng. "Old password")	
B4. User enters the new password into text box labelled "Nowe haslo"(Eng. "New password")	
B5. User enters the new password into text box labelled "Potwierdz haslo"(Eng. "Confirm password")	
B6. User clicks the button "Zmien" (Eng. "Change")	B7. System shows the message that the password has been changed successfully




## Appendix F – Critical evaluation by the client



Śląska Szkoła Jakości

Szkoła Podstawowa nr 22  
im. Juliusza Słowackiego w Rybniku  
44 – 270 Rybnik, ul. Boczna 17



Rybnik, 18/04/2008

### Critical evolution of the project by client

I am satisfied with the PSONline Project by Maciej Lipinski.  
Although the final product has minor drawbacks, ex. it does not to display Polish letters in Mozilla Firefox, the main objectives and expectation toward the project has been fulfilled. When tested in Internet Explorer browser, the final product passed all acceptance test which were performed.

It seems appropriate to mention, that the project conduction and project management by the student has been assessed positively. The management of the project was effective and client-friendly. The client was informed regularly about the project progress and was listened carefully when describing the needs.

SZKOŁA PODSTAWOWA NR 22  
im. Juliusza Słowackiego  
44-270 Rybnik 17, ul. Boczna  
tel. 42-29-950, fax 42-29-965  
Regon 000728078, NIP 642-27-07-251

DYREKTOR SZKOŁY  
*mgr Marek Lipiński*

---

www.rybnik.pl/sp22 Tel. 0 (32) 42 29 950 Fax 0 (32) 42 29 965 e-mail: sp22rybnik@poczta.onet.pl