**BIT302**

**SOFTWARE ENGINEERING**

**Assignment 1**

**Requirements**

**“Web-based Information System for MicroHousing System in Kuala Lumpur”**

Team Leader: Rivaldo Bagus Soepardhy (E1700882)

Contact Details: aldobagus@hotmail.co.id

Member: Luh Wulandari Maharani (E1700873)

Contact Details: [luhwulandari@gmail.com](mailto:luhwulandari@gmail.com)

**Table of Contents**

[Functional Requirements 3](#_Toc498518013)

[Non – functional Requirements 3](#_Toc498518014)

[Use Case Diagram 4](#_Toc498518015)

[Use Cases 5](#_Toc498518016)

[Expanded Use Cases 7](#_Toc498518017)

[Analysis Class Diagram 18](#_Toc498518018)

[Tasks Division 19](#_Toc498518019)

# Functional Requirements

**User Requirements:**

1. Clear login menu with user ID and password column so that students can access the system without confusion, that will lead the students to student home page
2. Forget password option if a student forgot his/her password
3. The system should have menu with buttons/icons that students can choose from, e.g. a person icon that students can click to see his/her personal information
4. Button to change login password
5. Clear logout button for the student to exit the application

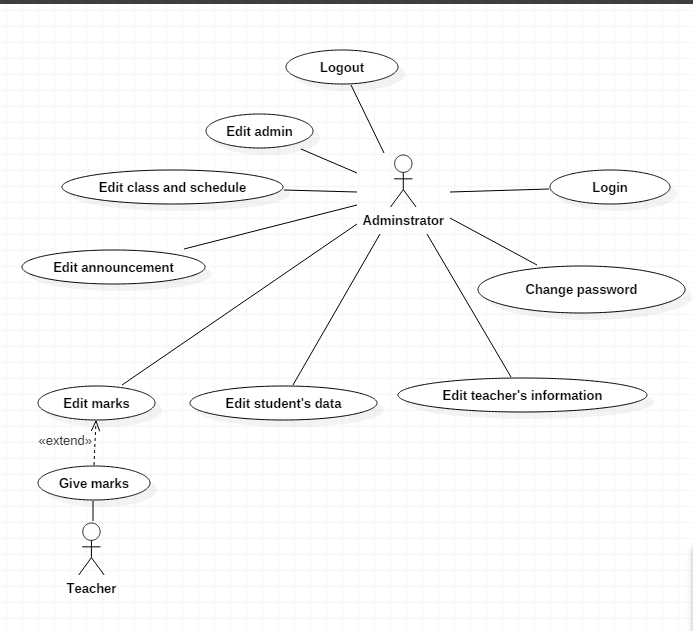
**Admin Requirements:**

1. Clear login menu with user ID and password column for admin to access the admin home page
2. Button to change login password
3. Admin page should have ‘edit menu’ to change data about students, teachers, marks and schedules
4. Admin should have ‘add menu’ to add new admin, students, teachers, marks, class and schedules
5. Admin should have ‘delete menu’ to delete an admin, student, teacher, announcement, mark and class
6. Menu to create announcement as a form that will be displayed in student home page

# Non – functional Requirements

1. Security requirement: using MD5 encryption for the password to make sure if someone without authorization can access our database, he/she can’t understand the password
2. Usability: the system should be easy to operate by admin and the features understood by students
3. Accuracy and precision: data entered by admin should be accurate and precise and free from error
4. Modifiability: data can only be modified by admin through admin page and students are only allowed to change his/her own password

# Use Case Diagram



# Use Cases

|  |  |  |
| --- | --- | --- |
| Requirement | Actor | Use Case |
| Login page has been created and functioned to allow admin and students to log in. | Student  Admin | Log in |
| Student and admin have been registered to the system and have had a login password | Student  Admin | Change password |
| The system provides a page for students to view their personal information and contact details | Student | View identity |
| The system has a page that shows contact details of all teachers in the school that students can view | Student | View teacher information |
| The system has a page that shows schedule based on the class of the student | Student | View schedule |
| A page for student to see their marks for home works, monthly tests, middle test, and final exam | Student | Check marks |
| Announcement that has been posted by admin will appear in student’s home page | Student | Check announcement |
| Teacher has given homework or test to students, and admin will input the marks to the system | Admin  Teacher | Edit marks |
| Students that have enrolled to the school will provide their personal data and contact details, and the information will be recorded in the system by admin | Admin | Edit student’s data |
| Information about teachers in the school will be recorded in the system by admin | Admin | Edit teacher’s data |
| Form to create announcement and it will be posted to student home page | Admin | Edit announcement |
| If there is a new class at the beginning of semester, new class and schedule will be created  If the class is already existed, the schedule will be updated at the beginning of the semester  If the class will no longer be needed, the class and schedule will be deleted at the beginning of the semester | Admin | Edit class and schedule |
| The system provides form to add or delete admin | Admin | Edit admin |
| The system provides log out button that will direct admin or student back to the login page | Admin  Student | Log out |

# Expanded Use Cases

1. Log in

|  |  |
| --- | --- |
| Use Case | Log in |
| Goal in Context | Allow admin and students to access the home page |
| Primary Actor  Secondary Actor | Admin and students  **-** |
| Typical Course of Events | |
| Actor Actions | **System Response** |
| 1. The process begins when the student or admin enter their user ID and password, and presses the login button | 2. The system checks the information received |
|  | 3. The system displays the home page for the registered user or for admin. |
| Alternative Course | |
| Line 3: If the student or admin entered wrong ID or password, the system will show alert message to tell if the information being entered is incorrect and student or admin should enter their information again. | |

2. Change password

|  |  |
| --- | --- |
| Use Case | Change password |
| Goal in Context | Allow admin and students to change their login password |
| Primary Actor  Secondary Actor | Admin and students  **-** |
| Typical Course of Events | |
| Actor Actions | **System Response** |
| 1. The process begins when the student or admin has been logged in |  |
| 2. Admin or student select the change password menu | 3. The system displays change password page with form to fill in the new password |
| 4. Admin or student has to fill in the form and type in the new password twice as validation |  |
| 5. Admin or student select the submit button | 6. The system will save the changes and update the database |
| Alternative Course | |
| Line 3: If two passwords being entered do not match, alert message will be shown and the new password cannot be submitted | |

3. View identity

|  |  |
| --- | --- |
| Use Case | View identity |
| Goal in Context | Allow students to access see their own personal information and contact details |
| Primary Actor  Secondary Actor | Students  **-** |
| Typical Course of Events | |
| Actor Actions | **System Response** |
| 1. Student has already accessed the home page and choose the personal information menu | 2. The system displays the personal information page |
| Alternative Course | |
| - | |

4. View teacher information

|  |  |
| --- | --- |
| Use Case | View teacher information |
| Goal in Context | Allow students to access teacher information page |
| Primary Actor  Secondary Actor | Students  **-** |
| Typical Course of Events | |
| Actor Actions | **System Response** |
| 1. Student has already accessed the home page and choose the teacher information menu | 2. The system displays the teacher information page |
| Alternative Course | |
| - | |

5. View schedule

|  |  |
| --- | --- |
| Use Case | View schedule |
| Goal in Context | Allow students to access schedule page |
| Primary Actor  Secondary Actor | Students  **-** |
| Typical Course of Events | |
| Actor Actions | **System Response** |
| 1. Student has already accessed the home page and choose the schedule menu | 2. The system displays the schedule page |
| Alternative Course | |
| - | |

6. Check marks

|  |  |
| --- | --- |
| Use Case | Check marks |
| Goal in Context | Allow students to check marks for home works or tests |
| Primary Actor  Secondary Actor | Students  **-** |
| Typical Course of Events | |
| Actor Actions | **System Response** |
| 1. Student has already accessed the home page and choose the marks menu | 2. The system displays the marks page |
| Alternative Course | |
| - | |

7. Check announcement

|  |  |
| --- | --- |
| Use Case | Check announcement |
| Goal in Context | Allow students to check announcement |
| Primary Actor  Secondary Actor | Students  **-** |
| Typical Course of Events | |
| Actor Actions | **System Response** |
| 1. Student has already accessed the home page and choose the announcement menu | 2. The system displays the newest announcement posted by admin |
| Alternative Course | |
| - | |

8. Edit marks

|  |  |
| --- | --- |
| Use Case | Edit marks |
| Goal in Context | Admin can input marks for students or edit the marks |
| Primary Actor  Secondary Actor | Admin  Teacher |
| Typical Course of Events | |
| Actor Actions | **System Response** |
| 1. Teacher has checked a home work or test of students |  |
| 2. Admin select to the marks menu in the system | 3. The system displays the marks page |
| 4. Admin add new marks for student based on the subject and the type of the mark (homework or test or exam) | 5. The system saved the changes and update the information that will be displayed in the marks page |
| Alternative Course | |
| Line 4: Admin can choose to edit existing marks in the system | |

9. Edit student’s data

|  |  |
| --- | --- |
| Use Case | Edit student’s data |
| Goal in Context | Admin can input information of new students or edit information of existing students |
| Primary Actor  Secondary Actor | Admin  - |
| Typical Course of Events | |
| Actor Actions | **System Response** |
| 1. Admin receives information about new students |  |
| 2. Admin select to the student menu in the system | 3. The system displays the student page |
| 4. Admin add new student to the database and his/her personal information and contact details | 5. The system saved the changes and update the information that will be displayed in the student page |
| Alternative Course | |
| Line 1: Admin receive information about existing student that want to change his/her data  Line 4: Admin edit the existing student’s data and update the database  Line 1: Admin receive information about student that will quit from the school  Line 4: Admin delete the student’s data and update the database | |

10. Edit teacher’s data

|  |  |
| --- | --- |
| Use Case | Edit teacher’s data |
| Goal in Context | Admin can input information of new teachers or edit information of existing teachers |
| Primary Actor  Secondary Actor | Admin  - |
| Typical Course of Events | |
| Actor Actions | **System Response** |
| 1. Admin receives information about new teachers |  |
| 2. Admin select to the teacher menu in the system | 3. The system displays the teacher page |
| 4. Admin add new teacher to the database and his/her contact details | 5. The system saved the changes and update the information that will be displayed in the teacher page |
| Alternative Course | |
| Line 1: Admin receive information about existing teacher that want to change his/her data  Line 4: Admin edit the existing teacher’s data and update the database  Line 1: Admin receive information about teacher that will quit from the school  Line 4: Admin delete the teacher’s data and update the database | |

11. Edit announcement

|  |  |
| --- | --- |
| Use Case | Edit announcement |
| Goal in Context | Admin can input new announcement or edit an existing one |
| Primary Actor  Secondary Actor | Admin  - |
| Typical Course of Events | |
| Actor Actions | **System Response** |
| 1. Admin receives information about new announcement |  |
| 2. Admin select to the announcement menu in the system | 3. The system displays the announcement page with form that admin can fill in |
| 4. Admin fill in the form and press the submit button | 5. The system will post the announcement to the student home page and admin announcement page |
| Alternative Course | |
| Line 1: Admin receives information to change some information from announcement that previously has been posted  Line 4: Admin delete the existing announcement before fill in the form and create a new one | |

12. Edit schedule

|  |  |
| --- | --- |
| Use Case | Edit class and schedule |
| Goal in Context | Admin can input new schedule at the start of the semester or edit the schedule if there is mistake when adding |
| Primary Actor  Secondary Actor | Admin  - |
| Typical Course of Events | |
| Actor Actions | **System Response** |
| 1. Admin receives information about new schedule for existing class |  |
| 2. Admin select to the schedule menu in the system | 3. The system displays the schedule page |
| 4. Admin add new schedule based on the class | 5. The system saved the changes and update the information that will be displayed in the schedule page |
| Alternative Course | |
| Line 1: Admin receives information that a new class is to be created with its schedule  Line 4: Admin add new class and add new schedule based on the new class  Line 1: Admin receives information to delete a class and its schedule  Line 4: Admin delete the class and its schedule | |

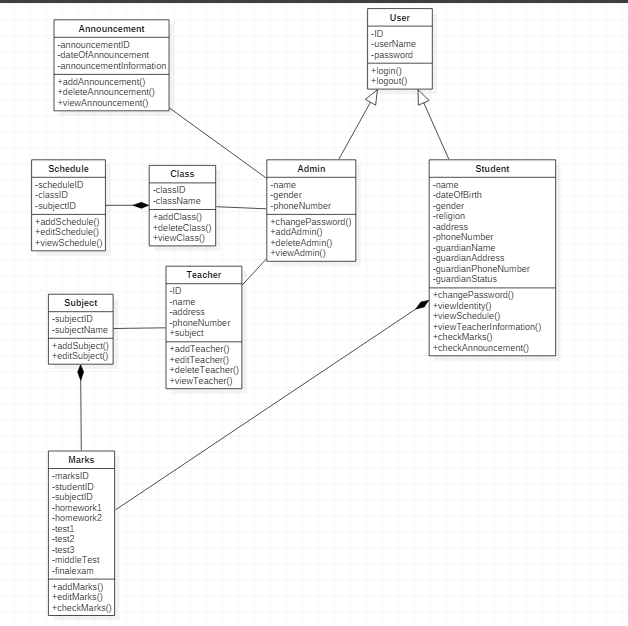
13. Edit admin

|  |  |
| --- | --- |
| Use Case | Edit admin |
| Goal in Context | New admin can be added to the system or delete existing one |
| Primary Actor  Secondary Actor | Admin  - |
| Typical Course of Events | |
| Actor Actions | **System Response** |
| 1. Existing admin receives information about new admin |  |
| 2. Admin select the add admin menu in the system | 3. The system displays the new admin page |
| 4. Admin add new admin to the database and his/her ID and password | 5. The system saved the changes and update the information that will be displayed in the existing admin page |
| Alternative Course | |
| Line 1: Admin receive information about admin that will quit from the school  Line 4: Admin delete the admin’s data and update the database | |

14. Log out

|  |  |
| --- | --- |
| Use Case | Log out |
| Goal in Context | Allow admin and students to exit the home page |
| Primary Actor  Secondary Actor | Admin and students  **-** |
| Typical Course of Events | |
| Actor Actions | **System Response** |
| 1. The process begins when the student or admin have finished what they need to do with the system |  |
| 2. Student or admin select the log out button | 3. The system displays login page |
| Alternative Course | |
| - | |

# Analysis Class Diagram



# Tasks Division

|  |  |
| --- | --- |
| Required Behaviour | Member’s Name |
| HTML and CSS script for designing the website menu and navigation | Lovely |
| PHP script for login | Lovely |
| PHP script for add new admin, delete and admin | Lovely |
| PHP script to change password | Denny |
| HTML and CSS script to design how data will be displayed when being viewed | Denny |
| PHP script for edit student, teacher, marks, class and schedules | Lovely and Denny |
| PHP script for search column in admin home page | Lovely |
| PHP script for posting announcement from admin to student home page | Denny |