Course Schedule Planner

Requirements Specification and Analysis

Version 1.0

10.03.2019

Berke Kağan Yıldız

Burak Sönmez



**Table of Contents**

Table of Contents

[REQUIREMENTS ANALYSIS DOCUMENT[1] 2](#_Toc529700747)

[1. Introduction 2](#_Toc529700748)

[1.1.Purpose of the System 2](#_Toc529700749)

[1.2.Scope of the System 3](#_Toc529700750)

[1.3.Objectives and Success Criteria of the Project 3](#_Toc529700751)

[1.4.Definitions, Acronyms, and Abbreviations 3](#_Toc529700752)

[1.5.Overview 4](#_Toc529700753)

[2. Current System 5](#_Toc529700754)

[3. Proposed System 5](#_Toc529700755)

[3.1.Overview 6](#_Toc529700756)

[3.2.Functional Requirements 6](#_Toc529700757)

[3.3.Nonfunctional Requirements 6](#_Toc529700758)

[3.4.System Models 7](#_Toc529700759)

[4.Glossary 20](#_Toc529700768)

[5. References 21](#_Toc529700769)

# REQUIREMENTS ANALYSIS DOCUMENT[1]

## 1. Introduction

The purpose of this system is to provide students with an easy way of being able to select their courses with their desires and showing best plans for them to select between them.

## 1.1.Purpose of the System

The driving reason behind developing this system is to give students an easy interface to see their classes and select them. Also automatically planning weekly schedules up to their choose of days, credit hours and wanted courses. More times than not, students try to take courses without visualising their weekly scheedules and they try to select and see their schedule step by step which took a lot of time and they only can do it in the course selection day, which makes them not to have time for planning, or they need to plan it before the day they choose courses with a paper and pen. It also shows most taken courses by students who are doing same major with them, so it would be easy for them to imagine which course to take with a statistic of chosen rate. There are varied input types for students to make the autmated schedule for them, so they can just select their requests and the program can automatically show them the best results.

## 1.2.Scope of the System

Our system will allow the students to make a weekly course schedule with their desires. The user first needs to sign up, then after logging in the system will list the courses he/she could take by categorising them. Students would select courses they wanted or make an automatic schedule with or without desired inputs.

Our system will only have student user, and by regirestering the system ask them to upload their transcript so the system would show the courses he/she would take and make a schedule by the student’s conditions.

The students would also see the chosen percentage of elective courses by the students in same major with them.

## 1.3.Objectives and Success Criteria of the Project

Our objective is to give our students the peace of mind and assurance of knowing how they will make their program and the varied of schedules to choose. By doing this we also want to save their time and avoid the fault that they could make by planning their program. And especially for the students who are not really knowing the rules of course selecting so they could try and see their conditions and this would make advisors to not be in a rush with their students in the selection day.

## 1.4.Definitions, Acronyms, and Abbreviations

* RAD: Requirement Analysis Document.
* Student: Only actor that the system is developed for.
* System: the software that is to be developed.

## 1.5.Overview

This RAD shall compare the current system that is being used which shall be replaced with the to-be developed system. It shall look at the upsides and benefits of replacing the old system. It also contains the system models which include functional and nonfunctional requirements which shall be described in detail and explained, there is also a use case description and scenarios that were created.

The organizational structure of the RAD is designed from the ground up so, we describe the things included from the basic descriptions of our system to the more complex system models and functionalities of the system. So we started with Introduction where we specify and familiarize the reader with what the project is all about and what it will do. The purpose of the system and the objectives of the system and what it is expected to achieve. Furthermore, we give an overview of the project which tells about what this RAD contains.

Next up is the current system section where we talk about the system that is currently being used by a majority number of people and this is the system we intend to replace and ultimately improve. We mention the things that make this current system inefficient and how is it time consuming and leaves room for uncertainty such as visiting a restaurant and it is full and so people have to search for an another place.

In proposed system section we speak on how the to-be developed system shall be an improvement on the old one. We discuss the advantages and why it is the right step forward. We mentioned that the current system is time consuming and uncertain, in the proposed system, this problem is solved by allowing the customer to reserve a table/ space for a number of people. If you reserve the space, you are guaranteed a table and this saves time and makes it easier for people to choose where to go. The proposed system will also be made to be as user friendly as possible so that customers of all ages and computer literacy can use it with minimal difficulty.

In overview section, we discuss the functionalities of the system which is what the system is going to be capable of and what actions it supports.

In functional requirements section, we discussed how the different users of our system -the manager, admin and customer- interact with the system and their different levels of authority. We make mention of what the Restaurant Booking system is capable of doing and its features how the users have different but sometimes similar functions.

In non-functional requirements section, we talk about functions that are not necessarily defining the behaviour of our system but rather, they are criteria for how the system should be judged or rated. These criteria include: usability, performance, reliability or supportability, implementation, interface, legal and packaging. We shall discuss these in more detail in later sections of the RAD.

In system models section, we talk about the scenarios, use cases, object model and dynamic model of the system. We created scenarios for manager, admin and the customer as our actors. Each scenario discusses how the actor shall partake in the use case and achieve the task at hand. For the use cases we gave full descriptions including, entry conditions, exit conditions, flow of events etc. The use case is an action to be performed by the specified actor within the system.

Lastly, in the glossary section we defined terms and other words or phrases needed to fully understand the RAD with minimal fuss. It is a guide for the reader who gets lost or misunderstands something.

## 2. Current System

The current system doesn’t allow students to plan their schedules, so they need to do it before the selection they with the excel document of current open courses that given by the university. So this would lead students to make faulty selections because they couldn’t control the conditions without trying to approve their courses. And there are currently no system to make automated programs to students and there is no statistic for them to see about elective courses.

## 3. Proposed System

The system to be designed is going to be made for making weekly course schedules to students. A student signs up and upload the transcript. After the registiration the system list the courses of the students that he/she could take in the current semester. Student could choose the courses by himself/herself or the student could just click to a create button and the system shows the best program for him/her by the student’s condition. The create schedule button would always fill the student’s schedule even if the student already chose course or courses. So if the student already chose two courses and the maximum number of courses is 5, then the system will add 3 more courses that would work best with that program and student. Every change on the schedule will be shown in real time schedule table so by clicking courses and removing them in the schedule would allow student to see it in real time in the schedule table. There will be inputs for create schedule button. The student could select the day-offs, maximum or minimum credits that he/she want to have in the program, or the student just select the desire courses in the list of courses that he/she could take. Depends on all those requests when the student press the create button the system will fill the program upon those requests.

## 3.1.Overview

In this section, we describe the functional requirements of our system and how they are implemented and who implements them. We have three users in our system. First, let’s start with the Customer. The customer has access to the most functionalities and he is the one who is able to: Make Reservation, Rate Restaurant, Edit Profile, Search Restaurant, Change Reservation, Cancel Reservation. These steps all require a login action to be performed first meaning, a customer must be registered on the system before having these functionalities available. Our customer is able to reserve a restaurant through our system and to this he logs in, he chooses the restaurant of this choice and he selects the number of guests he wants to reserve for. He is shown the available times and he chooses the one that suits him the most. After this he completes his reservation.

## 3.2.Functional Requirements

In our system, our main functionalities is making a schedule which is done by the student. As we mentioned earlier the student is a user who is registered to the system. To make a weekly course schedule he/she logs in and chooses from a list of available courses and/or press the create schedule button. The student also could choose the inputs for create schedule which are credit hours and off-day selections. Our system will show fill and show the schedule dependts on that inputs and/or courses that selected by the student. We have included a transcript update input in the settings page so the student update the transcript in every semester.

## 3.3.Nonfunctional Requirements

*Usability-* We made our system very usable and to measure this, making a schedule could only takes one click. And this is the main functionality of our system. So just

about anyone with basic knowledge of technology can use our system.

*Reliability-* Our system is expected to run 100% of the time so that our users can be able to plan their upcoming schedules at any time.

*Performance-* System performance needs to be optimal, this includes showing available courses and creating automatic schedules etc.. So our users won’t need to wait too long for system to load.

*Supportability-* System needs to be modified easily without difficulties.  *Implementation-* System shall be developed on Django framework. This means the programming language will be mainly based python which is really fast for parsing excel and html inputs to show the available courses.

*Interface-* Our interface will be a web page with .

*Packaging-* No packaging requirements.

## 3.4.System Model

Here, we show the high level funmctionalities of our system and how the users of our system interact with the system. For example, one of the functionalities our system is designed to make automatic weekly course schedule for the students. To do this, firstly the student needs to create an account. The student subsequently logs in after successfully creating the account. After this he/she sees the list of available courses and schedule table. The student then chooses the courses or/and presses the create schedule button which lead him/her to see the schedule. For the object models we depict our classes using UML diagrams and show the relation between them. In the dynamic model, we showed interactions between objects within a use case using sequence diagrams and showed how state machines are used to show the behaviour of that single object.

**Scenario 1:**

**Scenario name: Log-in**

**Participant actor instances: Ali: Student**

The flow of events:

1. When Ali open the course schedule planner website, System starts and brings Login screen
2. The username and password fields are filled.He presses the login button.
3. Both password and username match, there is no error so logging in process is successfully completed.

4.Ali click logout button and logout.

**Scenario 1 Extensions:**

2a. Ali inputs wrong username or password so logging in the process fails. He is asked to recheck the information he provided.

2b.Ali inputs wrong username or password so logging in the process fails. He is asked to recheck the information he provided. He realized he has forgotten his password. The system guides him to reset the password by sending a reset link to his attached email address.

5. Ali chooses no so next time he tries logging he has to manually input all his information.

**Scenario 2:**

**Scenario Name: Sign Up**

**Participant actor instances: Ali: Student**

The flow of events:

1-) Ali wants to use schedule planner For doing this, firstly he has to sign up the website. When he opening the website shows him Login and Sign-Up button. When he chose the sign-up button, system redirect him to the sign up page.

2-) The sign-up page consists of required information so he fills the username and password information about himself also he must upload his transcript.

3-) After click the sign-up button if any error occurs the system give him an information about it and want to fix them.

4-)System redirects him from sign up page to main page.

**Scenario 3:**

**Scenario name: Logout**

**Participant actor instances: Aysel: Student**

Flow of events:

1. Aysel has just completed schedule. She checks her profile.
2. She has finished doing all she wants to do and decides to log off.
3. She looks for the log out button in main page and click it.
4. System immediately logs him out.

5.Aysel logout.

**Scenario 4:**

**Scenario name: Update Transcript**

**Participant actor instances: Aysel: Student**

Flow of events:

1.Aysel wants to update her transcript.

2.She clicked settings button and redirected settings page.

3.She click update transcript button and add new transcript.

4.She click turn back main page button and redirect main page .

5.She logout

**Scenario 5:**

**Scenario name:Create Course Schedule**

**Participant actor instances: Ayşe: Student**

1.Ayşe wants to create course schedule so she logged in the System redirect him to main page.

2.She click Create Schedule button .

3.System create and shows best course schedule program for her.

4.She click logout buton.

**Scenario 5 Extensions:**

2a. She can enter input how many credits she wants to take.

2b.She can choose which day will be off.

2c.She can see available courses list and choose courses which she wants to take. Then, System create for her best schedule that include courses which she wants.

**Scenario 6:**

**Scenario name:Select Courses**

**Participant actor instances: Ayşe: Student**

1.Ayşe wants to create course schedule so she logged in the System redirect him to main page.

2. She sees the available courses and selection statistics for every elective courses by other students which are in same department with her.

3.She clicks on course which she wants and system add that course in to the schedule

4.She can create her course schedule with click on the create schedule button.

5.She click logout buton.

***Use case model***

**Use Case 1:**

|  |
| --- |
| *Use case name:* Login (Low Priority) |
| *Participant actors:* Student |
| *Flow of events:*  *1 )* The student first enters to Course Schedule Planner website.   1. The system presents the “Login Form” to the user. 2. The student enters username into username text field on the screen, also enters own password into password text field on the screen. Lastly, the user sends a request to the system by using login button on the screen to be logged in.   4. The system checks the username and password from the “Person” table so that the system allows the user to login. |
| *Entry Condition:* The student enters to login screen. |
| *Exit Condition:* The student is logged in, OR,The user has received an explanation indicating why he/she could not login. |

**Use case 2:**

|  |
| --- |
| Use Case Name: Sign Up(Low Priority) |
| Participant Actors: Student |
| Flow of Events:  1-) The person is firstly open the website and see the sign up button and presses it.  2-) opens the sign up page.  3-) The person is filling username, password , confirm password and upload transcript.  4-)The person presses the sign up button and it will send a request to the system. |
| Entry Condition: The person enters to the sign up screen. |
| Exit Condition: The user signed up OR The system has send an error message about why he/she couldn’t signed up(can be invalid username/password). |

**Use case 3:**

|  |
| --- |
| *Use case name:* Logout (Low Priority) |
| *Participant actors:* Student |
| *Flow of events:*  1)The person click on log out button |
| *Entry Condition:* The student logged in system. |
| *Exit Condition:* The student logout system. |

**Use case 4:**

|  |
| --- |
| *Use case name:* Update Transcript(High Priority) |
| *Participant actors:* Student |
| *Flow of events:*  1)The person click update Transcript button.  2) Upload new transcript html document. |
| *Entry Condition:* The student logged in system.Then,open settings Page. |
| *Exit Condition:* The student logout system. |

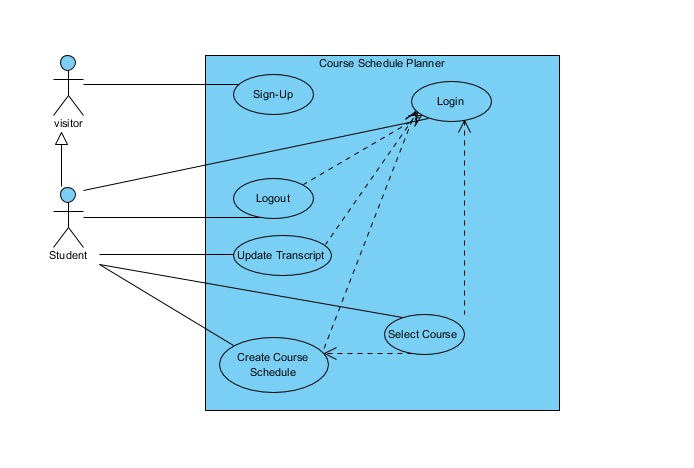
**Use case 5:**

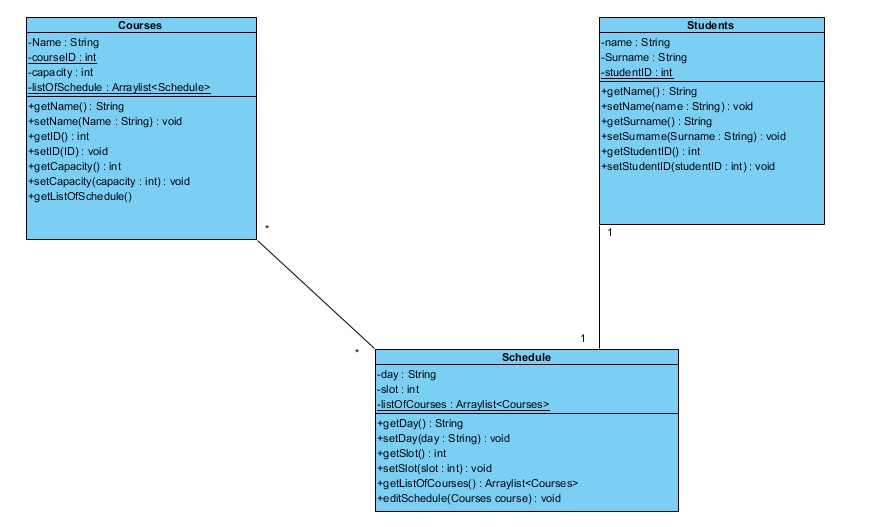
|  |
| --- |
| *Use case name:* Create Course Schedule (High Priority) |
| *Participant actors:* Student |
| *Flow of events:*  1)The student can select which day will be off and course which she wants to take .  2)The student click create schedule.  3) System create best available course schedule for her. |
| *Entry Condition:* The student logged in system. |
| *Exit Condition:* The student logout system. |

**Use case 6:**

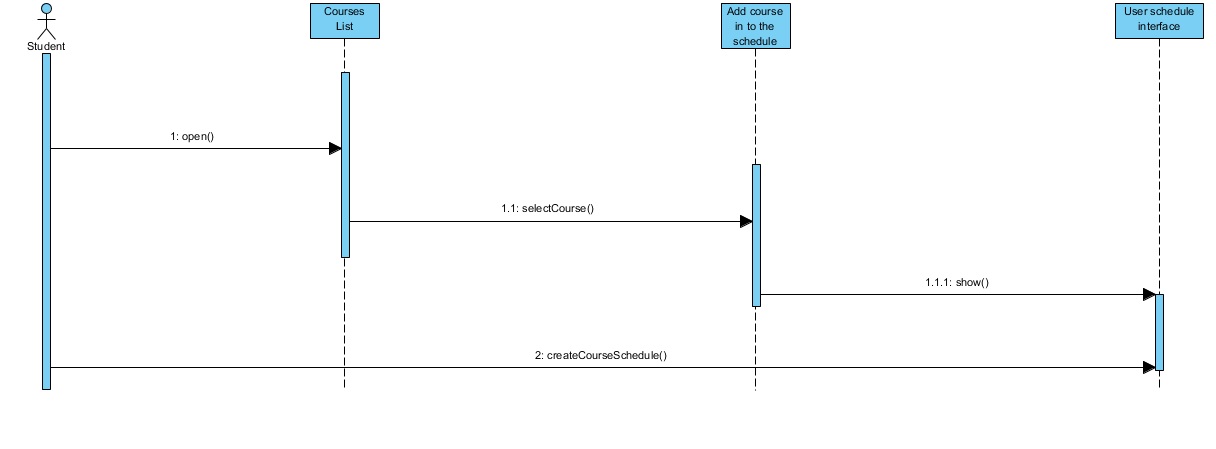
|  |
| --- |
| *Use case name:* Select Courses (High Priority) |
| *Participant actors:* Student |
| *Flow of events:*  1)The student can see selection statistics for elective courses by other students who are in the same department.  2)The student click on the course which she wants to take.  3)System add that course in schedule. |
| *Entry Condition:* The student logged in system. |
| *Exit Condition:* The student logout system. |

*Use Case model*

*Object model*

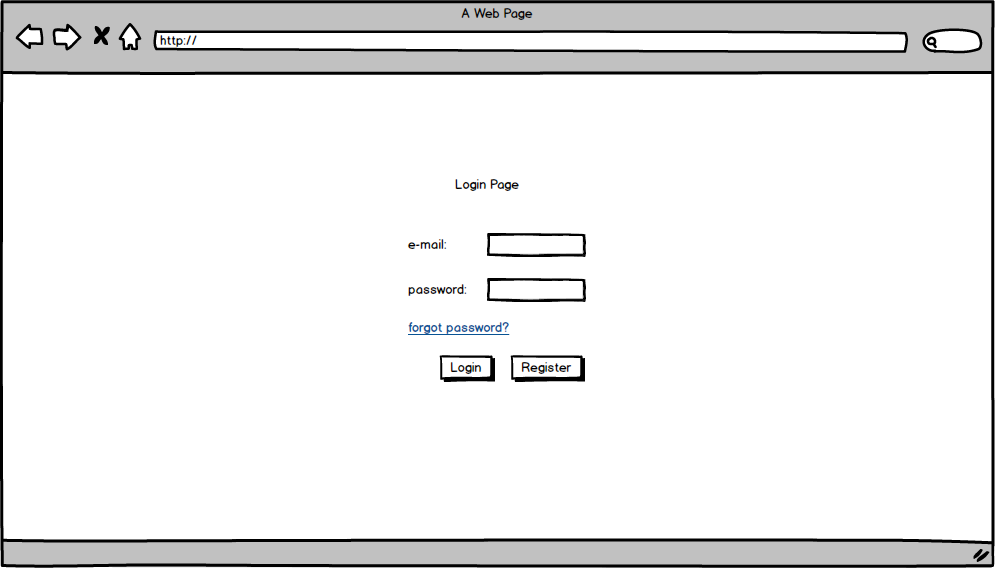


*Dynamic model*

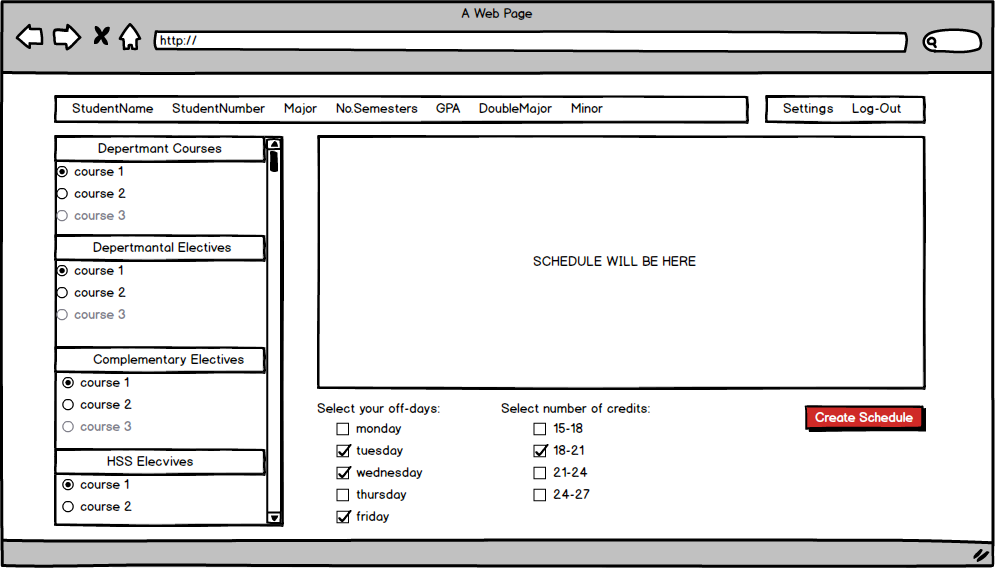


**Mock-Ups**

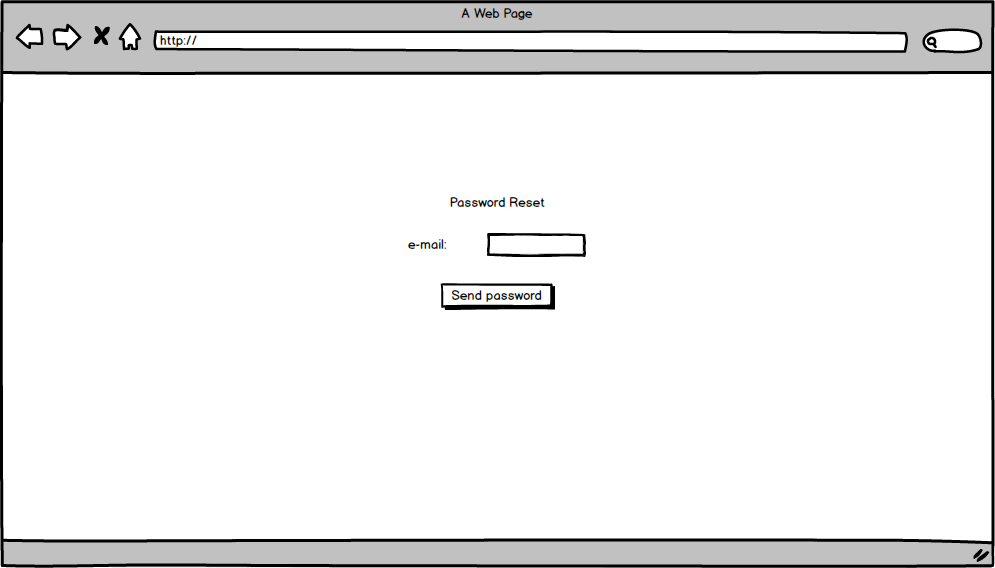
Login



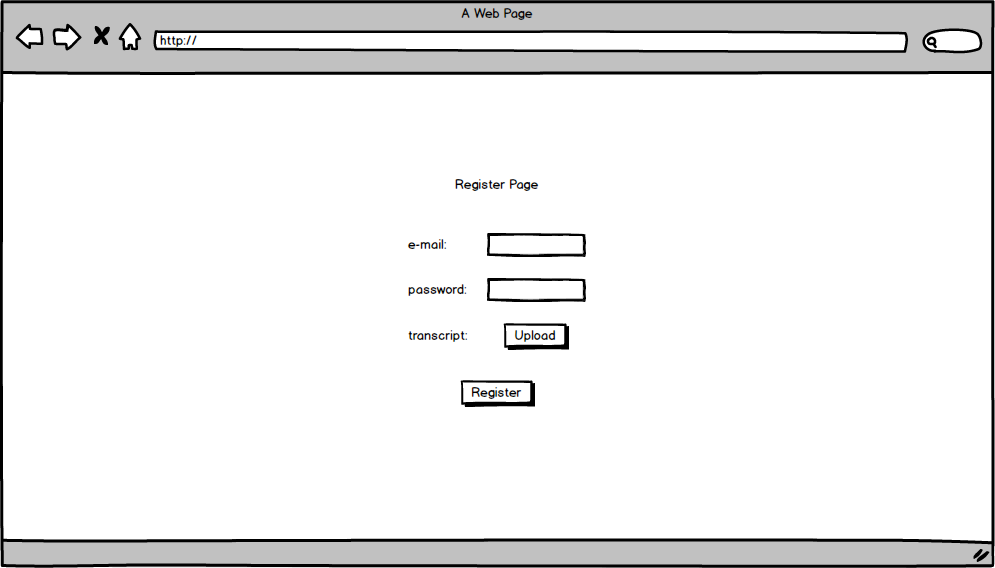
Main Page



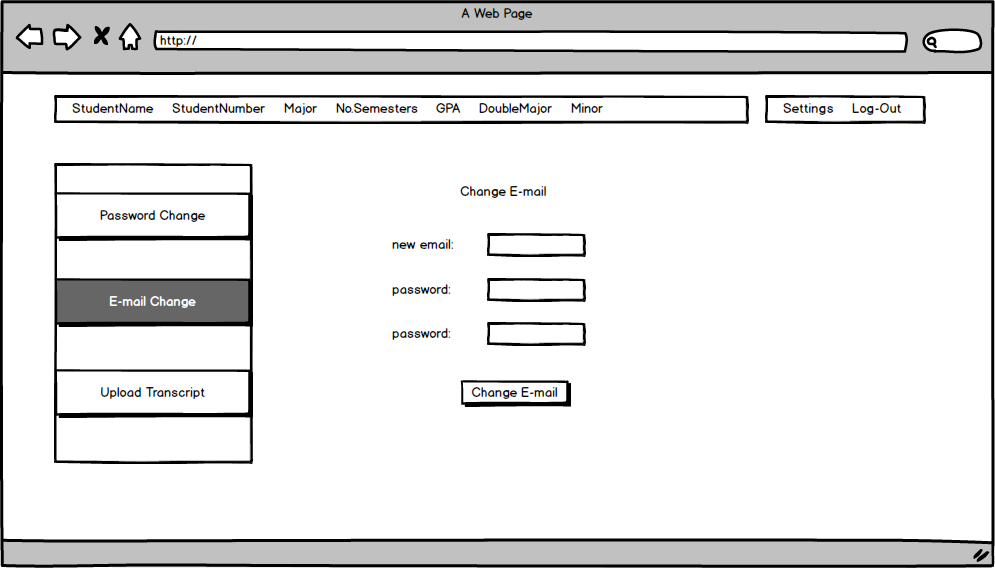
Password Reset



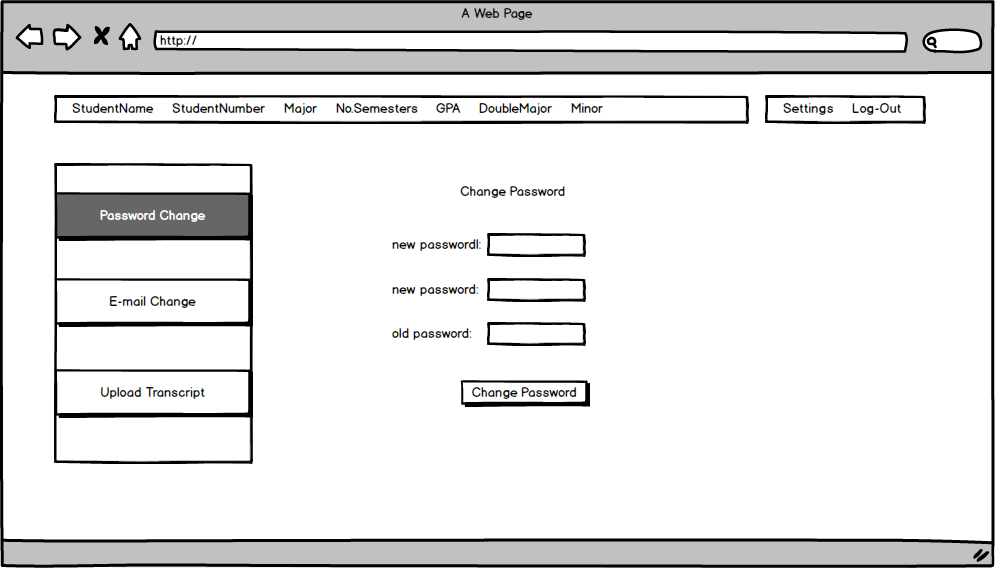
Sign Up



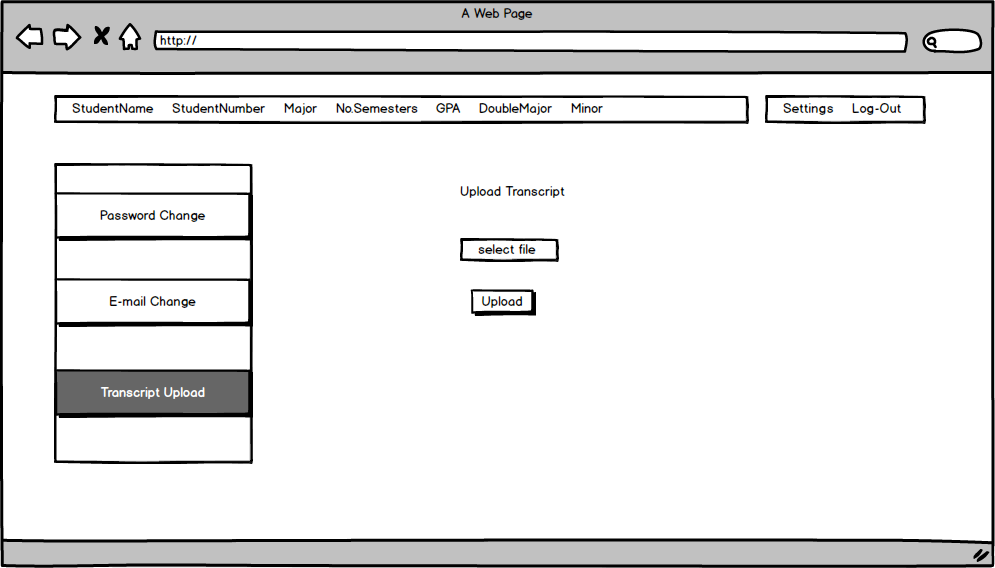
Settings e-mail



Settings password



Settings Transcript



## 4.Glossary

Log in: Being entered to the Course Schedule Planner System with a specific registered account.

Log out: Being exited from the the Course Schedule Planner System with a specific registered account.

**F**unctionality: An action or control supported by our system for any of the users to carry out a task.

Password: A secret word or phrase that must be choose of all students when they are register.

Profile: A section where student's informations.

Sign Up: A section for visitors to be a student with entering their specific informations.

Student:The person who use the course schedule planner.Create a course schedule.

Use Case: A methodology used in system analysis to identify, clarify, and organize system requirements.

Visitor: A person who is an unregistered user in the system.

## 5. References

1. Bruegge B. & Dutoit A.H.. (2010). *Object-Oriented Software Engineering Using UML, Patterns, and Java*, Prentice Hall, 3rd ed.