Software Requirements Document for [ ]

TEAM: \_\_\_\_\_\_\_

AUTHORS: \_\_\_\_\_\_\_\_\_\_\_

1. REWRITE SECTIONS IN THIS DOCUMENT WITH YOUR OWN DESCRIPTIONS
2. OMIT SECTIONS MARKED OMIT

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Change** |
| 0.1 |  | SM |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1 Introduction 3

1.1 Purpose 3

1.2 Scope 3

1.3 Definitions, acronymns, abbreviations 3

1.4 References 3

1.5 Overview 3

2 Overall Description 4

2.1 Product Perspective 4

2.2 Product functions 6

2.3 User characteristics 19

2.4 Constraints 19

2.5 Assumptions and Dependencies 19

3 Specific Requirements 20

3.1 External Interface Requirements 20

3.2 FEATURES 20

3.3 Performance requirements 20

3.4 Design Constraints 20

3.5 Software System Attributes 20

3.6 Other Requirements 21

Appendix 22

# Introduction

## Purpose

The purpose of this document is to establish how the application should interact with the end user, and establish all application requirements functional, and non functional. Once finalized, this document will state what must be accomplished for the application to be considered finished.

## Scope

This SRS covers a number of potential use cases that users may encounter, as well as an overview of the project and its intended uses. It also includes information on the project’s UI sketches, but the primary purpose is to give detailed descriptions of anticipated use cases.

## Definitions, acronymns, abbreviations

|  |  |
| --- | --- |
| Term | Description |
| 4-year Plan / Flowchart/Schedule | A student’s plan for which classes he or she will be taking in the upcoming years. The 4-year plan is displayed in a flowchart format to make it easier to read. The terms “4-year Plan” and “flowchart” are used interchangeably throughout the documentation. |
| Course / Class | A Course offered regularly at Iowa State University. The course must be offered during at lest one of the following terms: fall, spring, or summer. Courses may or may not have prerequisite courses, and may or may not have a credit value. |
| Prerequisite | A specific class that is required to be complete before a student may in role in a class more difficult class. |
| Template | A pre-made 4-year plan that fulfills all degree requirements and provides a balanced and engaging course schedule. |

## References

## Overview

[OMIT]

# Overall Description

Students at Iowa State are required to produce a 4-year study plan during their freshman year. The process of doing that is long and difficult; this application aims to streamline the process, while also providing a way for advisers to view all of their advisees 4-year plans. Advisers will also be able to upload templates, or suggested 4-year plans that have worked for students in the past.

## Product Perspective

“Doodle”, a lightweight, easy to use scheduling web-app

“appointy”, a scheduling app for small businesses

“My Class Schedule: TimeTable”, a mobile app used to schedule classes and other appointments

“Degree Planner: GPA Calculator”, a mobile app used to track you GPA over the course of college

### Concept of Operations

This project will be a web-based application that allows the user to create an account on a webpage, design a schedule for themselves, and save that schedule. If the user is an adviser, that adviser will be able to add advisees to their account, view their advisee’s schedules, and design templates for students to use. Administrators will have access to all user accounts, all schedules and templates, and all courses/degrees. They will be able to edit, add, and remove all of these as they wish.

This system will be supported by a database, which will store lists of users, classes, degrees, and other information. By dividing the database into a number of smaller pieces, we hope to streamline use and improve performance. Users will request data from the lists when creating schedules, making accounts, editing information, etc.

### Major User Interfaces

PUT SCREEN FLOW HERE

#### Example Screenshot and description

PUT EACH SCREEN SKETCH HERE.

### Hardware Interfaces

Any device that supports a web browser and current standards of HTML, CSS, etc

### Software Interfaces

// example: CGI-URL or function signatures etc (OMIT for now).

### Communication Interfaces

// example: modem etc (OMIT for now)

### Memory Constraints

// RAM, and other storage constraints (OMIT for now)

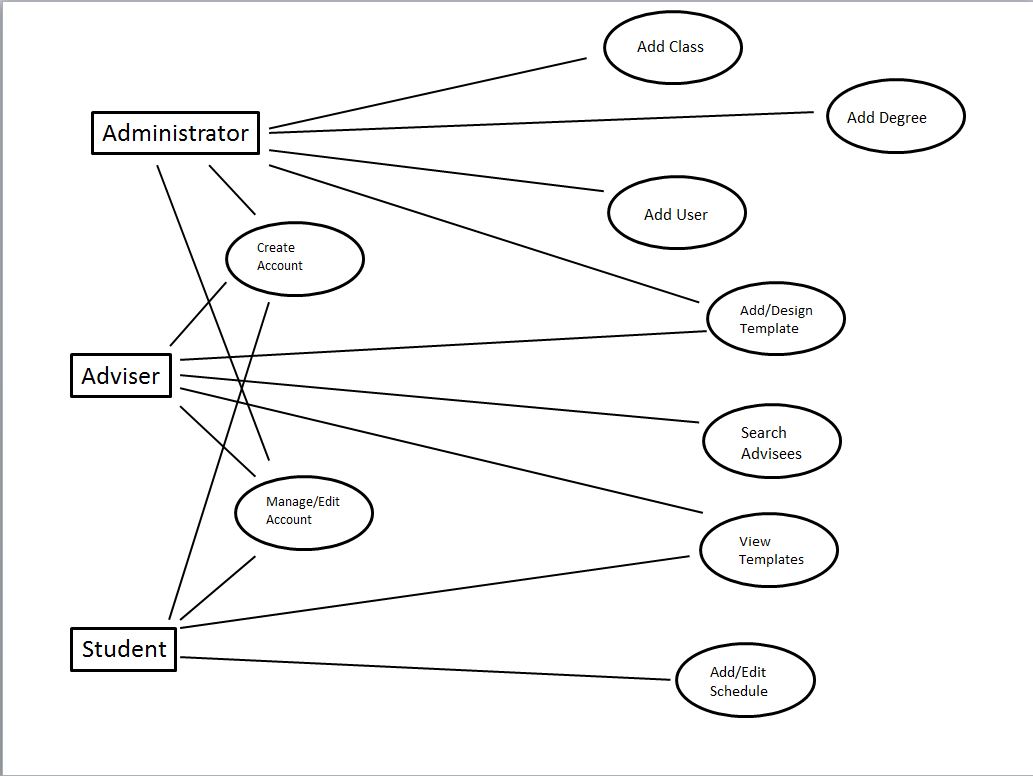
### Operations

// special operations (if any) (OMIT for now)

### Site Adaptation Requirements

//ex: Japanese language etc (OMIT for now)

## Product functions



These use cases describe the steps that a student, adviser, or administrator would need to take in order to accomplish some common/basic tasks, such as creating an account or designing a new 4-year plan.

EITHER USER STORIES OR USECASES ARE FINE IN THE SUBSECTIONS. CAN BE SHORT DESCRIPTIONS \_ NOT AS ELABORATE AS BELOW

### UC-0

Student Adds or Edits a 4 year plan (User goal / Student)

*Main scenario:*

1. Student goes to the applications website.
2. The system displays log in page.
3. The student enters log in credentials and then hits log in.
4. The system opens the information about the student and displays the student’s home page.
5. If student is adding a flow chart, Student selects the add Flow chart option.
6. The system displays a template selector screen.
7. Student selects a major for the new 4 year plan.
8. Student may then either brows the list of templates available, or select the blank 4 year plan option.
9. The system then opens the requested template or blank 4 year plan in the 4-year plan editor.
10. The student now adds or removes classes as he or she desires.
11. The student saves his or her changes by clicking the save and close button.
12. The system will save the changes to the flow chart and display the student’s home screen.
13. The student clicks the log out button in the upper right hand corner.
14. The computer will log the student out and display the sign in page.

*Extensions:*

3a. Student attempts to log in, but no account for the user exists.

3a1. The student instead selects to create an account.

3a2. The system brings up the create user account page.

3a3. The student fills the form in and selects submit.

3a4. The computer takes this information and creates an account for the user and then displays to them a blank user home page.

3a5. The user can now select to add a new flow chart and continue on with what they intended to do.

6a. Student wishes to edit an existing schedule.

6a1. The student selects the flowchart that he or she wishes to modify and then clicks edit.

6a2. The system opens the selected plan in the 4-year plan editor.

7a. Student whishes to have a double major

7a1. The student clicks on the + sign by the major field.

7a2. The system adds a second major field.

7a3. The student fills out the second major field

7b. The student whishes to have a minor.

7b1. The student fills the field labeled minor.

2.2.2 UC-1

Nicholas Riesen

“Look up Student Information” (User Goal / Adviser or Administrator)

*Main scenario:*

1. User opens applications web site.
2. The system displays log in page.
3. User enters credentials and hits log in.
4. System looks up the user’s account and shows them the home screen
5. In the left hand side of the screen, there is a search student’s panel. The user would select a major, and or type in a name and press search.
6. The system will update the list of students shown in the panel according to the search terms.
7. The user will select the student he wishes to view.
8. The system will display the student information in the center panel.
9. After viewing the information the user wants, the user presses the log out button in the upper right hand corner of the screen.
10. The system logs the user out and displays the beginning log in information.

*Extensions:*

3a. The users log in information is not recognized.

3a1. The user will contact a different administrator who will create an account for him.

3a2. The user will then continue as he or she originally intended.

5a. The student is not enrolled in the system.

5a1. Adviser contacts the student and tells him or her she must enroll in the system.

5a2. The student makes an account in the system.

5a3. The adviser can now see the students information.

5b. The student is not one of the Advisers Advisees.

5b1. The adviser can contact the advisor of the student and request the information.

5b2. The advisor can contact an administrator and request that the information be changed.

### 2.2.3 UC-2

Haoyu Liu

"Add/Design Template" (User goal / Administrator)

Main scenario:

1. Administrator search courses by the characteristic(Course Name, Category)
2. Administrator select course and view the relevant information of the course.
3. Administrator add courses into the template.
4. Administrator submits the template for verification by the system.
5. The system verifies the template if it is available to storage.
6. The system stores the relevant information about the template into the database, thus the template has been created within the system.

Extensions:

4a. The template is not in correct format.

4a1. System will display a message that details the error. Example (“The total credits is not enough, need more courses). The system will then redirect to the add/design screen with the original template.

5a. User enters information that cannot be stored. (Example: template already exist)

2a1. System will display a pop-up message on and occurrence of this event. It will detail the problem associated with the event and redirect the administrator back to the add/design template page.

### 2.2.4 UC-3

Haoyu Liu

"Create Account" (User goal / All users)

Main scenario:

1. User enters the relevant information (Name, User name, Email address and Password).
2. User submits the information for verification by the system.
3. The system verifies that the information provided is in the correct format for storage.
4. The system stores the relevant information into the database, thus the application has been sent to the administrator.

Extensions:

2a. User enters information that is not in the correct format

1a1. System will display a message that details the error. Example (“The password has to be longer than 8 digit and contains both letters and numbers”). The system will then redirect to the create account page with the correct information still within their relevant text frames.

2b. User does not enter in all the necessary information.

2b1. The system will display a message detailing the problem, such as “The email address is needed”. It will then redirect back to the create account page with the correct information.

3a. User enters information that cannot be stored. (Example: Email address already in use)

3a1. System will display a pop-up message on and occurrence of this event. It will detail the problem associated with the event and redirect to the create account page with other information is stored. The system will clear the box input where the duplicate information was entered, leaving room for the user to try again.

### 2.2.5 UC-4

Jake Caithamer

Edit/Manage User Account (User goal / All Users)

Main Scenario:

1. A user wishes to alter their profile information
2. If they aren’t already logged in, the user will need to log in by visiting the Scheduling Assistant homepage and logging in using their credentials, which brings them to their homepage
3. User will click Manage Account in the top right of their homepage to open the Account screen
4. Account screen will display the user’s name, profile picture, position, and bio, along with any schedules associated with this user
5. User makes desired changes and presses Save Changes to return to their homepage

Extensions

1. User wishes to change their password
   1. From the account page, the user presses Change Password, which opens a new page
   2. This page will prompt the user to enter their current password
   3. If the entered password is correct, system will then prompt the user to enter their new password twice
   4. If these two new passwords match, then the system displays “Password Change Successful” and returns to the Account screen.
2. User wishes to update their profile picture
   1. From the account page, the user presses Upload New Profile Picture
   2. User selects from images on their hard drive
      1. Selected image can be at largest 400x400 pixels
   3. The selected image is uploaded to the database, replacing the current profile picture
3. Administrator wishes to change someone else’s account information
   1. Administrator searches the desired individual from the administrator homepage
   2. After finding them, the administrator presses Edit Account
   3. This opens a page similar to the Account page, where the administrator can alter any of the selected user’s information, including password, name, etc
   4. After making changes, the administrator presses Save, which returns them to the administrator homepage

2.2.6 UC-5

Jake Caithamer

Adviser Add a Template Schedule (User goal / Adviser)

Main Scenario:

1. A school’s adviser identifies an area of study in which a complete range of template schedules are not available
   1. Caused by events such as a new Major being added to the curriculum, or by a student suggesting a new combination of majors/minors in their field of study
2. Adviser logs into the Schedule Assistant system using their credentials. They will be brought to the adviser homepage
3. Once at their homepage, the adviser selects or creates a schedule to add as a template
4. The adviser gives the schedule a descriptive title and detailed description
5. Adviser saves the schedule as a template, making it available for students to consult when planning their 4-year plan

Extensions:

1. Adviser adds a new template
   1. Adviser selects an existing schedule.
      1. From the adviser homepage, an adviser can browse a list of their advisees and their schedules
      2. Adviser may select one of these schedules and select Add as Template
      3. This will prompt the adviser to alter the title and description of that schedule
      4. Once the title/description are updated, the adviser selects Save As Template to add it to the database and make it available for students to access
   2. Adviser creates a new schedule
      1. Adviser selects Design New Template from the adviser homepage
      2. This opens the Schedule Designer
      3. The schedule designer prompts the adviser for major(s) and minor(s), after which the adviser may add classes to the new template
      4. Once the adviser is happy with the class structure, they may add a title and description to the template
      5. At this point, the adviser presses Save As Template. This closes the Schedule Designer and saves the template to the database, making it available for students to access

2.2.7 UC-6

Nicholas Rabbitt

Add a Degree (User Goal / Administrator)

Main Scenario:

1. The demand for a new degree increases leading to the need to create one.
2. The administrator logs in, leading them to their homepage.
3. They then navigate to the Degree Editor page.
4. At this page they can click on the “Add” button to add a degree.
5. Then, the administrator can fill out the details for the degree.
6. Once finished, the administrator can save the information into the degree database, so that advisers and students can use them.

Extensions:

1. Administrator removes a degree.
   1. The administrator goes to the Degree Editor page.
   2. There is a search tool that they can use to navigate to the desired degree.
      1. This tool has the option to narrow a search to the name of the degree or the department it’s listed under.
   3. Once the degree is selected, the administrator then pushes the “Remove” button.
   4. Before deleting the information from the database, a prompt appears requesting confirmation of the attempt to delete.
   5. Once the administrator selects “OK” the degree is deleted from the database.
      1. If “Cancel is selected nothing happens.
2. Administrator edits a degree.
   1. The administrator goes to the Degree Editor page.
   2. They select the desired degree using the search tool.
   3. The “Edit” button is then pushed.
   4. This causes the details of the degree to populate the text boxes.
   5. The administrator can edit the details of the degree until it appears as they intend.
   6. Once finished, the administrator can choose to either save or cancel the changes.

2.2.8 UC-7

Nicholas Rabbitt

Add a Class (User Goal / Administrator)

Main Scenario:

1. The need arises to add a new class.
   1. This is caused by the increase in demand by students.
   2. Another cause could be the change in requirements for a certain degree.
2. The administrator logs into their respected accounts, leading them to the homepage.
3. Once there, they can navigate to the Class Editor page.
4. At the class editor page, the administrator can fill out the given information to make the class details accurate and easy to interpret.
5. After the class details appear correct, the class is saved to the classes database to allow students and advisers to include them in there schedule planning.

Extensions:

1. Administrator removes a class.
   1. Once in a while, a class gets ruled obsolete.
   2. From the administrator homepage, navigate to the Class Editor page.
   3. In this page, there is option to search for a class.
      1. This search can be narrowed down to the name of the class, the department, or category it lies in.
   4. Once a class is selected, the administrator can choose to delete the class.
   5. Before deleting from the database, there is a prompt to assure the system that the user meant to delete the class.
   6. If “OK” is selected the class data will be erased from the database.
      1. If “Cancel” is selected, nothing will change.
2. Administrator edits a class.
   1. Again the administrator navigates to the class editor page.
   2. The search tool allows the administrator to select the desired class.
   3. Once a class is selected, they push the “Edit” button.
   4. The blank details boxes are then populated by the information on the class.
   5. The administrator can edit that data, at that point.
   6. Once edited, there is the option to save the changes or cancel them.

2.2.9 UC-8

John Stuart

"Add User" (User goal / Administrator)

Main scenario:

1. Administrator receives application from prospective student.
2. Administrator enters student information into the relevant information frames.
3. Administrator submits the information for verification by the system.
4. The system verifies that the information provided is in the correct format for storage.
5. The system stores the relevant information into the database, thus “creating” the student account within the system.

Extensions:

1a. Administrator enters information that is not in the correct format

1a1. System will display a message that details the error. Example (“Wrong date format, please use MM/DD/YYYY). The system will then redirect to the add student screen with the correct information still within their relevant text frames.

1b. Administrator does not enter in all the necessary information.

1b1. System will display a message detailing the problem, such as a missing name. It will then redirect back to the add student page.

2a. User enters information that cannot be stored. (Example: Username already in use)

2a1. System will display a popup message on and occurrence of this event. It will detail the problem associated with the event and redirect the administrator back to the page where the information is stored. The system will clear the box input where the duplicate information was entered, leaving room for the administrator to try again.

2.2.10 UC-9

John Stuart

"Add Advisee" (User goal / Advisor)

Main scenario:

1. Advisor is given notice that they have been assigned a new student.
2. The Advisor retrieves the relevant information related to the student (Name, ISU ID, Net ID, Major).
3. The Advisor enters the relevant information into the add Advisee block, located on the Advisor Homepage.
4. The system verifies that the information provided is in the correct format for storage.
5. The system stores the relevant information into the database, thus “adding” the student as the adviser’s advisee within the system.

Extensions:

1a. Advisor enters information that is not in the correct format

1a1. System will display a message that details the error. Example (“Wrong NetID, please enter in the 9 digit code located on your ID). The system will then redirect to the advisor homepage with the correct information still within their relevant text frames.

1b. Advisor does not enter in all the necessary information.

1b1. The system will display a message detailing the problem, such as a missing name or ID. It will then redirect back to the advisor homepage.

2a. User enters information that cannot be stored. (Example: ID already in use)

2a1. System will display a popup message on and occurrence of this event. It will detail the problem associated with the event and redirect the advisor back to the page where the information is stored. The system will clear the box input where the duplicate information was entered, leaving room for the advisor to try again.

## User characteristics

// typical user characteristics, frequency of usage etc

## Constraints

// all conditions that may limit design options (INCLUDE NON FUNCTIONAL CONSTRAINTS)

## Assumptions and Dependencies

// hardware and software assumptions and dependencies

# Specific Requirements

// Here you need to put in details (if any). Mark items [None] if you do not have any information.

HERE instead of looking at users and user stories, look at features of the system.

For example, You can think of a car and view it in terms of features. For example, steering, cruise-control, air-bags, 4-wheel-drive etc.

DO ENTER NONFUNCTIONAL REQUIREMENTS (like maintainability, extensibility etc)

## External Interface Requirements [OMIT THIS SECTION]

### User Interfaces

### Hardware Interfaces

### Software Interfaces

### Communications Interfaces

## FEATURES

### FEATURE-1 ….

#### …

#### …..

## Performance requirements

## Design Constraints

## Software System Attributes (THESE ARE NON-FUNCTIONAL REQUIREMENTS)

### Reliability

### Availability

### Security

### Maintainability

### Portability

## Other Requirements

// ADD Appendices (if any)

// Regenerate Table of Contents