Software Requirements Specification and Design document

for

CSUSM Student App

Version 1.0 approved

Prepared by JJ Javier, Kenneth Wang, William Phong

JKW

3/29/2023

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
JJ J, Kenneth W, William P	3/29/2023	Initial Draft	1.0
JJ J, Kenneth W, William P	4/2/2023	Final Draft	1.0

1. Introduction

1.1 Purpose

The purpose is to create a mobile application that provides a streamlined way to access school related information. The application will provide students of CSUSM a way to monitor their school assignments and be delivered relevant information catered to their needs. Students and professors will be able to create accounts and receive notifications and other information related to their classes. This SRS will describe all of the software requirements and functions for the CSUSM Student Application. The document will describe necessary requirements and functions for the application, and provide visual representations through diagrams to allow for further understanding of the system.

1.2 Document Conventions

This document follows IEEE formatting conventions. The document is written in Arial, double spaced, and font sizes for main headers are 16 pt, and subsection headers and body sections are 12pt. Section headers and subsections are numbered and bolded. Every requirement statement has its own priority.

1.3 About our project team

William Phong CSUSM ID: 201347983, email: phong005@csusm.edu
JJ Javier CSUSM ID: 200790153, email: javie032@csusm.edu
Kenneth Wang CSUSM ID: 201031719, email: wang175@csusm.edu

JKW is a 3 man startup group based in California focused on improving student's day to day lives. We strive to provide and provide solutions that adapt to an individual's personal education needs by creating applications that help students use their time efficiently and provide creativity in balancing their everyday needs.

1.4 Intended Audience and Reading Suggestions

The intended audience of our proposal document are students of our SE370 class, the professor, administrators, and mentors of any kind. The SRS will also be utilized by the developers throughout the project. This SRS contains information pertaining to the top level requirements of the project, statement of work, timeline, and specific features of the product. The suggested reading order of this document is in order, with 6 main sections, and subsections underneath each one.

1.5 Product Scope

The purpose of this application is to provide a streamlined process to enhance students' day to day lives. An inefficient or bloated user interface leads to less usage of the software and ends up not providing any benefit to students. We want to optimize the user experience towards convenient and efficient purposes, allowing students to upbring their success rates while keeping track of their progress. This product aims to improve students' time management and prioritize each individual's personal needs.

2. Statement of Work

The JKW Group and CSUSM will work and communicate together in order to provide a product that meets the requirements provided.

2.1 Communication

Our team has come up with multiple solutions in order to maintain effective communication throughout our semester long project. In the case of any urgent messages such as difficulties/misunderstandings, changes or updates, clarifications, etc, we have a team Discord server as well as a messenger group chat. To address more specific and significant occurrences, the team will have weekly in person meetings in order to set our sights clear and make sure we stay on track to the schedule we commit to. We are also utilizing Github and Monday to update each other on our progress. The team, if necessary, will be able to text or email each other as well as the professor for any unclear concerns while also having each other's outlines and objectives for what we expect from each other.

2.2 Dependencies and Constraints

We will be utilizing Java for the frontend of the mobile application as well as implementing the backend. A MySQL database and connector will be used to establish a connection between the application and database. Dependencies and constraints may change as the project progresses and changes in requirements occur. There are no known constraints to the project at the time of creation.

2.3 Design, Development, and Implementation Methods

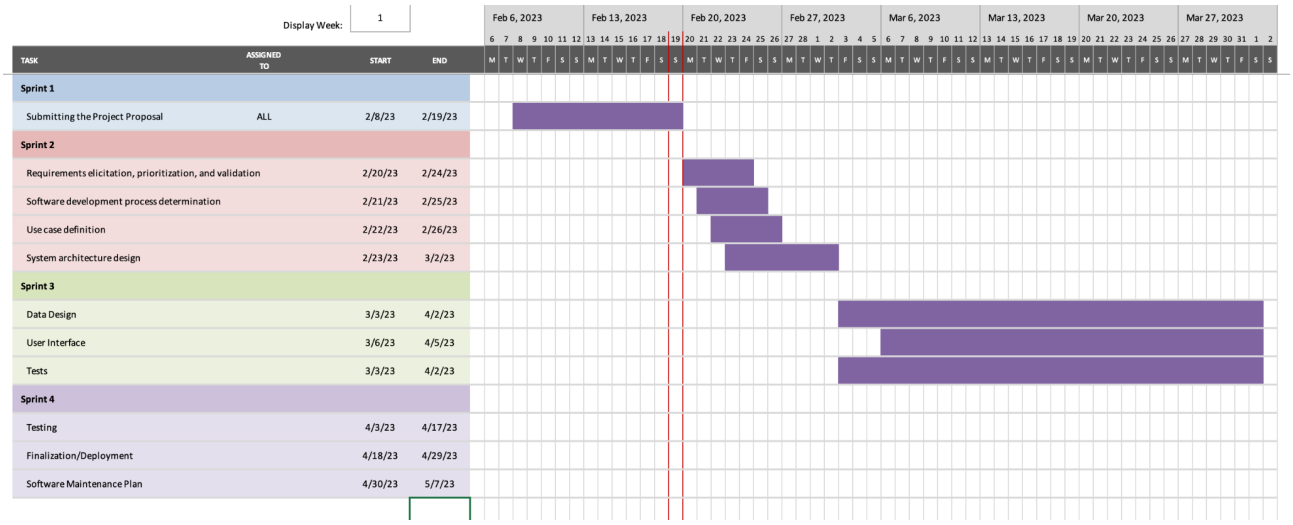
Intellij will be used for the Java implementations of the application. A MySQL database and connector will be used to establish a connection between the mainframe and database. GitHub will be used for the team to review and commit their implementations and work, as well as an online backup of code. The team will be using the agile model to pace the

progress of the application as a more optimal approach for the scope of our project. Scrum/Monday will be used to update each team member's progress and any issues that arise. For quality assurances, we will create tests and update them as we add onto the application.

2.4 Change Management

Changes will be communicated between the team members and acquirers to be evaluated and approved if certain criteria's needs to be changed. If the change is approved, the project's schedule and quality will be resolved through scrums. Communication between the members is of utmost importance to ensure that the expected developments are done on schedule. Team members will keep each other accountable for individual shortcomings/issues.

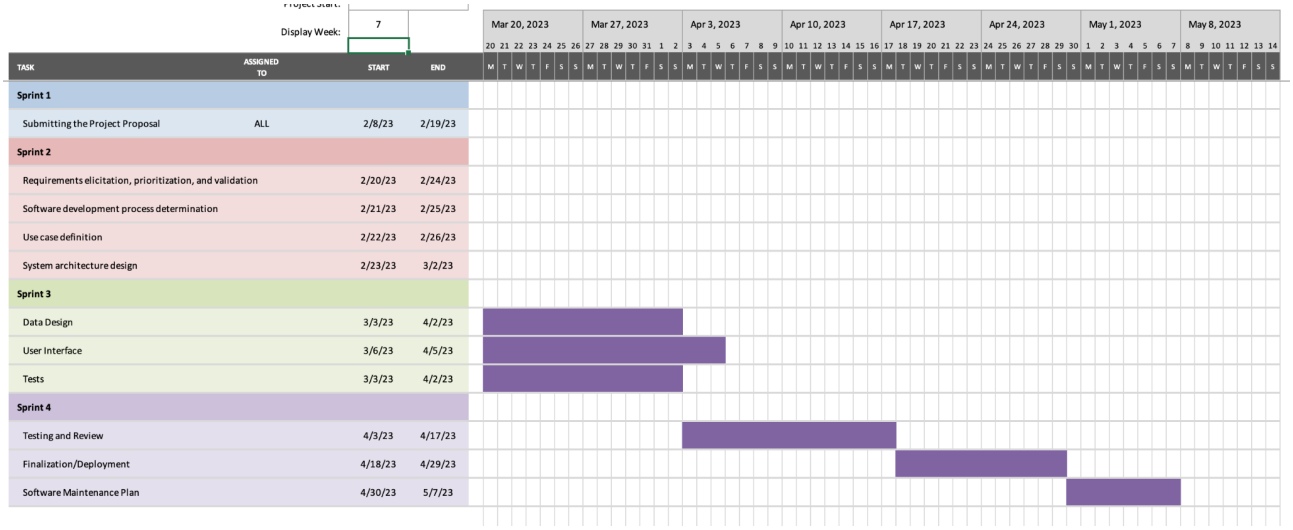
3. Timeline



TASK	ASSIGNED TO	START	END
Sprint 1			
Submitting the Project Proposal	ALL	2/8/23	2/19/23
Sprint 2			
Requirements elicitation, prioritization, and validation		2/20/23	2/24/23
Software development process determination		2/21/23	2/25/23
Use case definition		2/22/23	2/26/23
System architecture design		2/23/23	3/2/23
Sprint 3			
Data Design		3/3/23	4/2/23
User Interface		3/6/23	4/5/23
Tests		3/3/23	4/2/23
Sprint 4			
Testing		4/3/23	4/17/23
Finalization/Deployment		4/18/23	4/29/23
Software Maintenance Plan		4/30/23	5/7/23

Project Start:
Display Week: 7

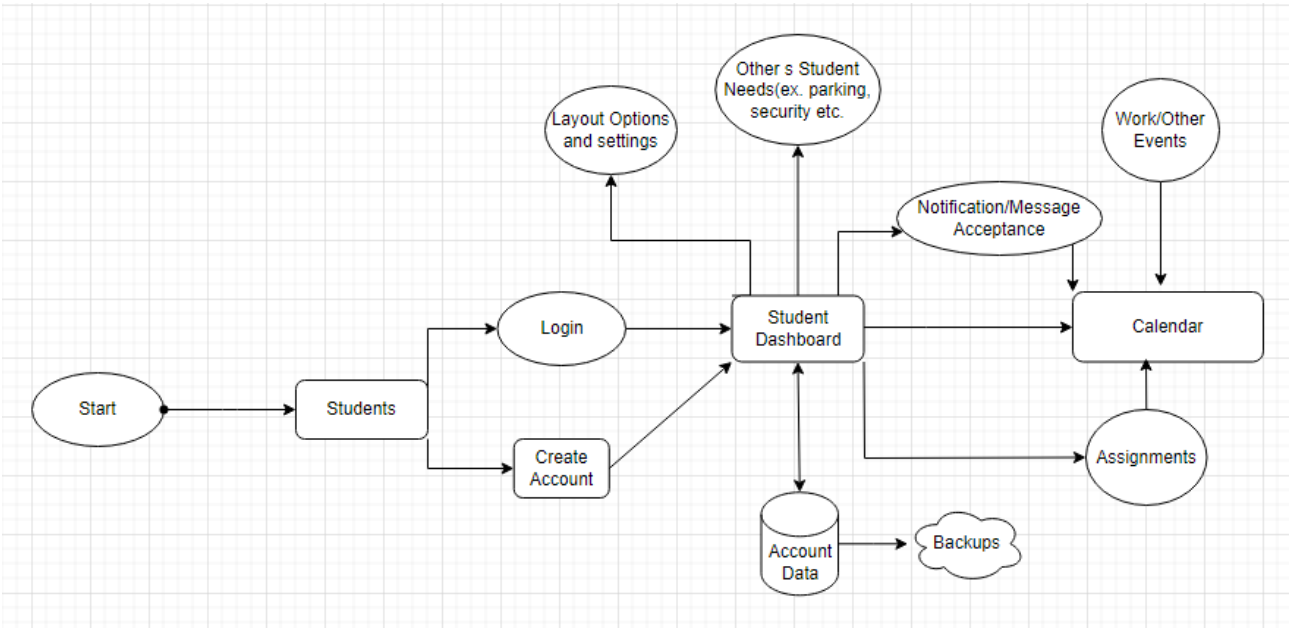
TASK	ASSIGNED TO	START	END
Sprint 1			
Submitting the Project Proposal	ALL	2/8/23	2/19/23
Sprint 2			
Requirements elicitation, prioritization, and validation		2/20/23	2/24/23
Software development process determination		2/21/23	2/25/23
Use case definition		2/22/23	2/26/23
System architecture design		2/23/23	3/2/23
Sprint 3			
Data Design		3/3/23	4/2/23
User Interface		3/6/23	4/5/23
Tests		3/3/23	4/2/23
Sprint 4			
Testing and Review		4/3/23	4/17/23
Finalization/Deployment		4/18/23	4/29/23
Software Maintenance Plan		4/30/23	5/7/23



4. Overall Description

4.1 Product Perspective

This application is made to improve and/or replace the existing CSUSM app.



4.2 Product Functions

- Create and log into a school account
- Let's users monitor their school priorities with a schedule(class and assignment) planner and notification system.
- Receive notifications

- Edit the layout of the application
- Allow users to frequently update their information

4.3 Technical requirements

The product must be written in Java code. The application will require a computer/server to host the database and communicate with the application. A student user will need to have a mobile device with internet access and sufficient memory space available to store the product and the data implemented.

4.4 Operating Environment

The application will operate on iOS and Android mobile devices. It will be written in Java and utilize a MySQL database and Java connector to interact with the database.

4.5 Design and Implementation Constraints

An implementation issue exists in regards to the login system. We are currently unsure of how to securely create a login system and our application is a proof of concept. Another constraint is with regards to utilizing Java and MySQL to create databases. iOS devices do not support Java applications so we need to find a way to port it to iOS. Underdeveloped skills in creating mobile applications.

4.6 User Documentation

The application will have an in-built faq/tutorial page to assist the user, as well as separate documentation of how the application functions.

4.7 Assumptions and Dependencies

The transition from Java to our desired mobile application, as well as the proper usage of a MySQL database, can affect our project progress as we are unfamiliar with their functions. Our user interface or database structure may also change due to new design or tested requirements. Our project at the moment has no external factors or third-party/commercial constraints.

5. External Interface Requirements

5.1 User Interfaces

The user interface will be implemented using tools and software packages in Java in Intellij. There will be UI pages where users can click on the classes and assignments they want to view.

5.2 Hardware Interfaces

The user must have an iOS or Android supported device in order to use the application, along with a stable internet connection. The application will not have a high hardware requirement so most modern devices will be supported.

5.3 Software Interfaces

The application will utilize the Java database connector to be able to connect to a MYSQL database that will consist of the information of the user (student, professor, or technician) and class/assignment data.

5.4 Communications Interfaces

This application will have an email function where they can email certain groups of people or professors in each class. This application should support all web browsers. The application will not be able to be of use without stable internet connection or any invalid email addresses.

6. System Features

System features will include a login system, create account feature, and assignment update feature.

6.1 Login

6.1.1 Description and Priority

Access to course materials.

Priority: High
Benefit: 9
Penalty: 1
Risk: 1
Cost: 1

6.1.2 Stimulus/Response Sequences

User Action: Opens the application and log in
System Response: System identifies the account to allow or deny access. If logged in, displays the dashboard for that account

User Action: Student clicks on a specific course.
System Response: App displays the assignments and dates of the assignment in that course.

User Action: Student clicks on an assignment.
System Response: App displays the assignment specifications. Also allows users to download the materials

6.1.3 Functional Requirements

REQ-1: The user must have an account and be able to log in using their username and password. If the information is wrong, give notice and let the user retry
REQ-2: The system should show all the classes that the student is enrolled in and allow them to check each class and assignment.

6.2 Assignment Update

6.2.1 Description and Priority
Assignment submission for students
Priority: High
Benefit: 9
Penalty: 1
Risk: 1
6.2.2 Stimulus/response Sequence
User Action: User clicks on an assignment.
System Response: Displays the assignment information showing due dates, submission status, grade and allows users to upload assignments to be submitted.

User Action: User upload file.
System Response: App verifies the file and shows a copy of the submission in the file format.

User Action: User submit file.
System response: Uploads the file to the server, records the submission and time it was submitted and updates the submission status.
6.2.3 Functional Requirements
REQ-1: Users should be able to click or have a touch screen device to click or press to input files.
REQ-2: The file should be of the type acceptable for that assignment.

6.3 Notification System

6.3.1 Description and Priority
Calendar and Reminder
Priority: High
Benefit: 9
Penalty: 1
Risk: 1
6.3.2 Stimulus/response Sequence
User Action: User navigates to calendar from dashboard

System Response: Displays the calendar view of events, exam dates, and assignment due dates.
User Action: User click on add event.
System Response: App shows an interface for students to add events, date, time, location and description.
User Action: User clock on save.
System Response: App validates the information save the new event and adds it to the calendar, if the event has conflict with another, gives a notice.
User Action: User clicks on create event.
System Response: App shows an interface for students to add reminder name, date, time, location, and description.
User Action: User clicks on save.
System Response: App validates the information, saves the new reminder onto the server and gives notification/reminder to the user when the saved date and time arrives with a description and location of the reminder.

7. Other Nonfunctional Requirements

7.1 Performance Requirements

The software should respond to user inputs, such as button clicks and upload files. The app should be able to save and update changes made by the users. The software should be available and usable to users 24/7, except while under scheduled maintenance. The software should be run smoothly across mobile devices and any browsers.

7.2 Safety Requirements

The software will follow the CCPA(California Consumer Privacy Act) to protect users personal information and ensure that the data collected and stored are protected securely. The data recorded will not be sent to any third party services.

7.3 Security Requirements

The software will have a secure socket layer to encrypt data transmitted between the user and the server. Without proper username and password specifications, the user won't be able to enter his/her profile.

7.4 Software Quality Attributes

7.4.1 Adaptability: Users will be able to customize or edit the layout of the application. New features will be added if necessary.
7.4.2 Availability: The application will be available to users 24/7.
7.4.3 Correctness: The application should meet its functional requirements.
7.4.4 Flexibility: Changes and modifications will be tested and ran over a testing model so the maintenance time of the application for changes can be minimized.

7.4.5 Interoperability:The software should be compatible with different platforms and devices.

7.4.6 Maintainability: The application should be easy to update and troubleshoot.

7.4.7 Portability: The application will be compatible with different devices and operating systems.

7.4.8 Reliability: The application should not crash under normal operation and repeated inputs.

7.4.9 Reusability: Parts of the codebase should be reusable for future add-ons.

7.4.10 Robustness: The software should be able to handle unexpected inputs and errors without causing a crash or data loss in the system.

7.4.11 Testability: Testings will be run using automated testers.

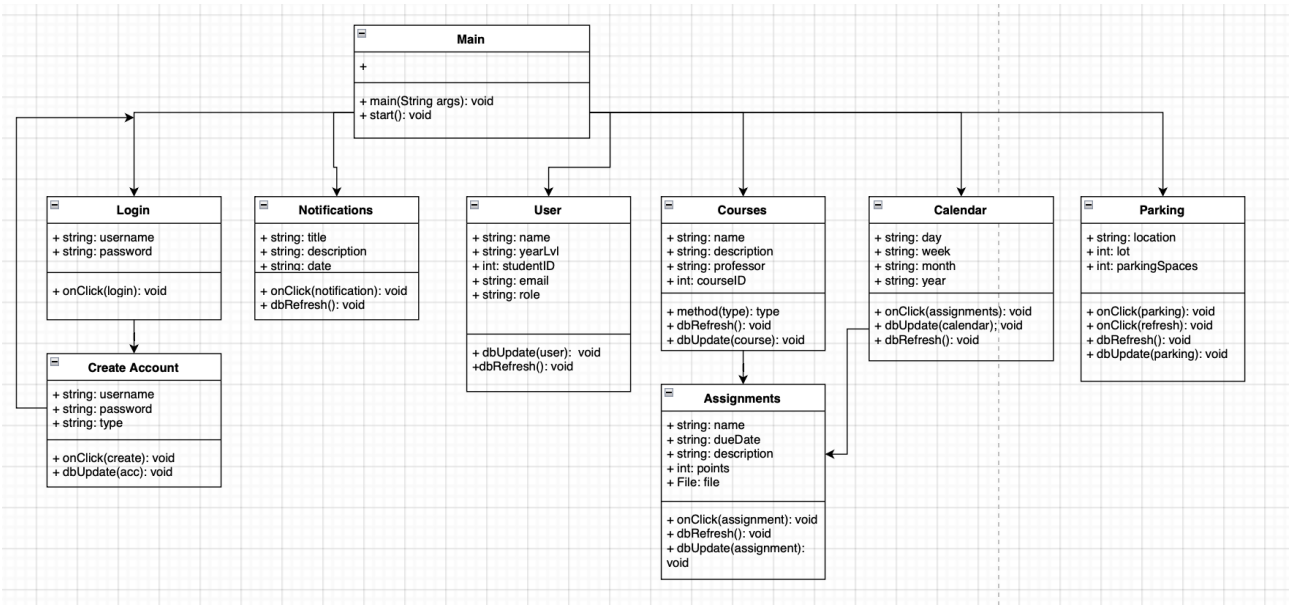
7.4.12 Usability: The software will be user friendly and easy to navigate.

7.5 Business Rules

Only teacher accounts will be able to create assignments, while students can only submit assignments within the range of deadlines set by the teachers. Assignments are not to be changed by the students, but only for them to access.

8. System Architecture

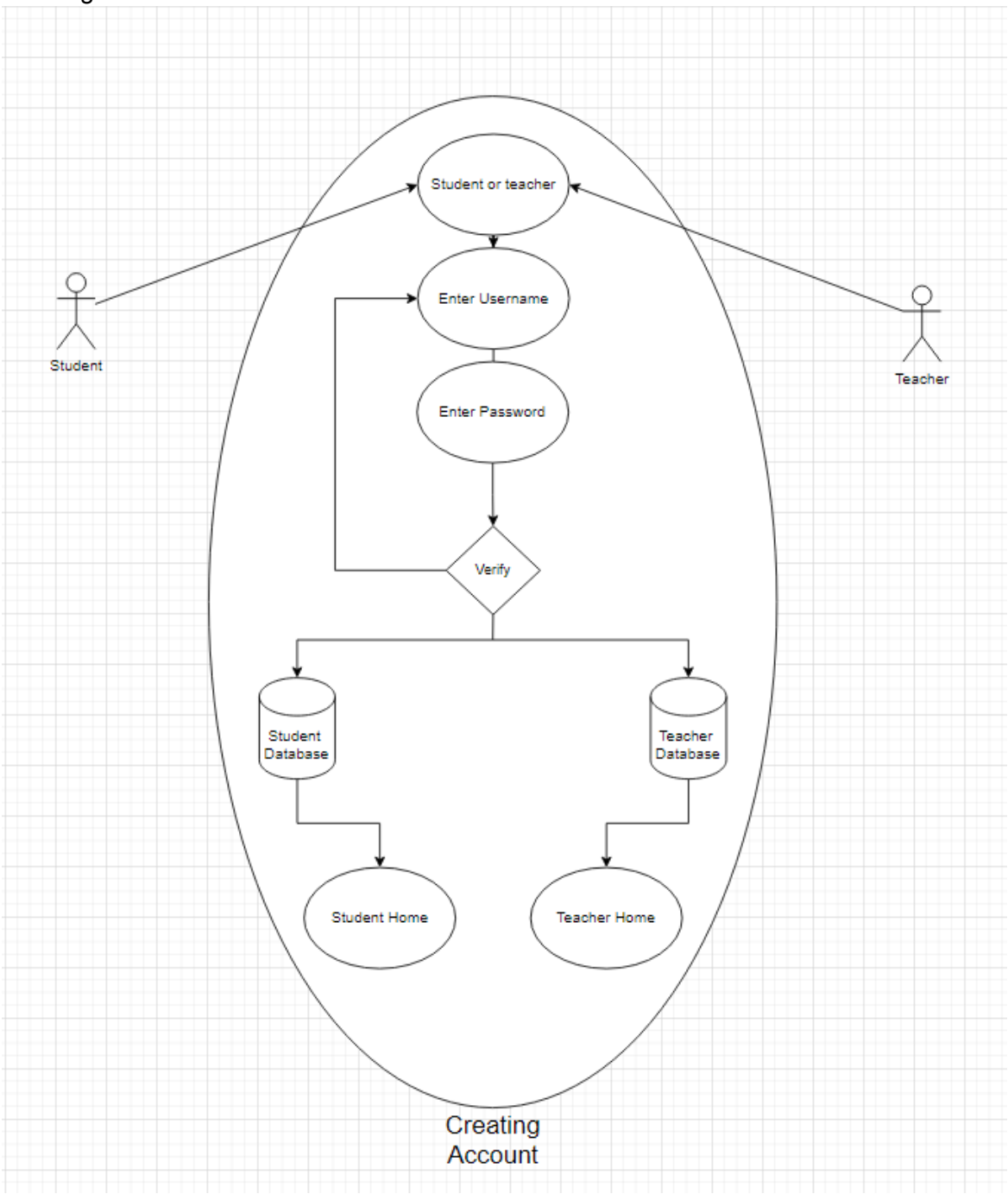
8.1 Architectural Design



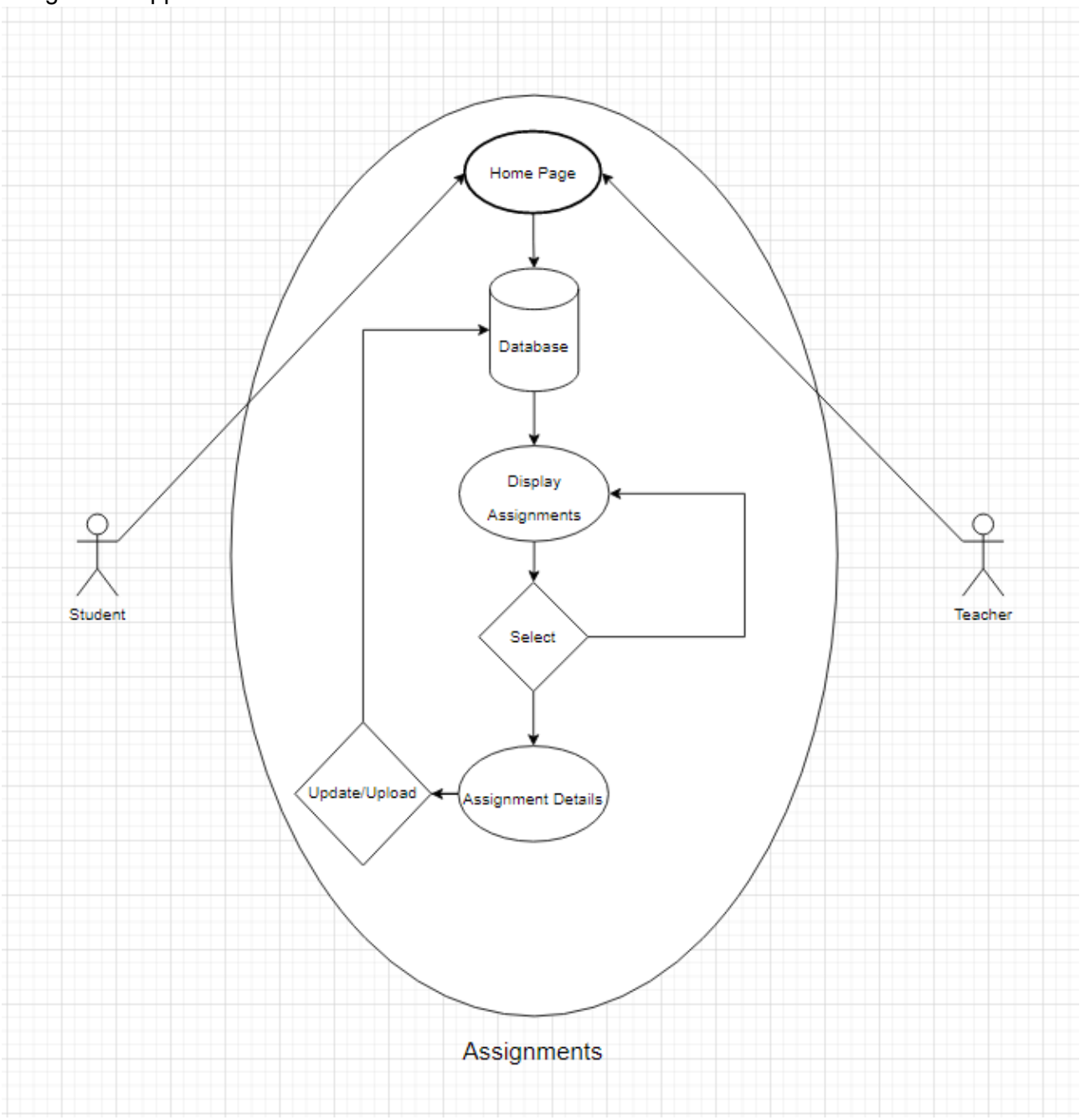
All modules will draw and update information to and from the database, allowing for each individual piece to always be constantly updated and work together

8.2 Decomposition Design

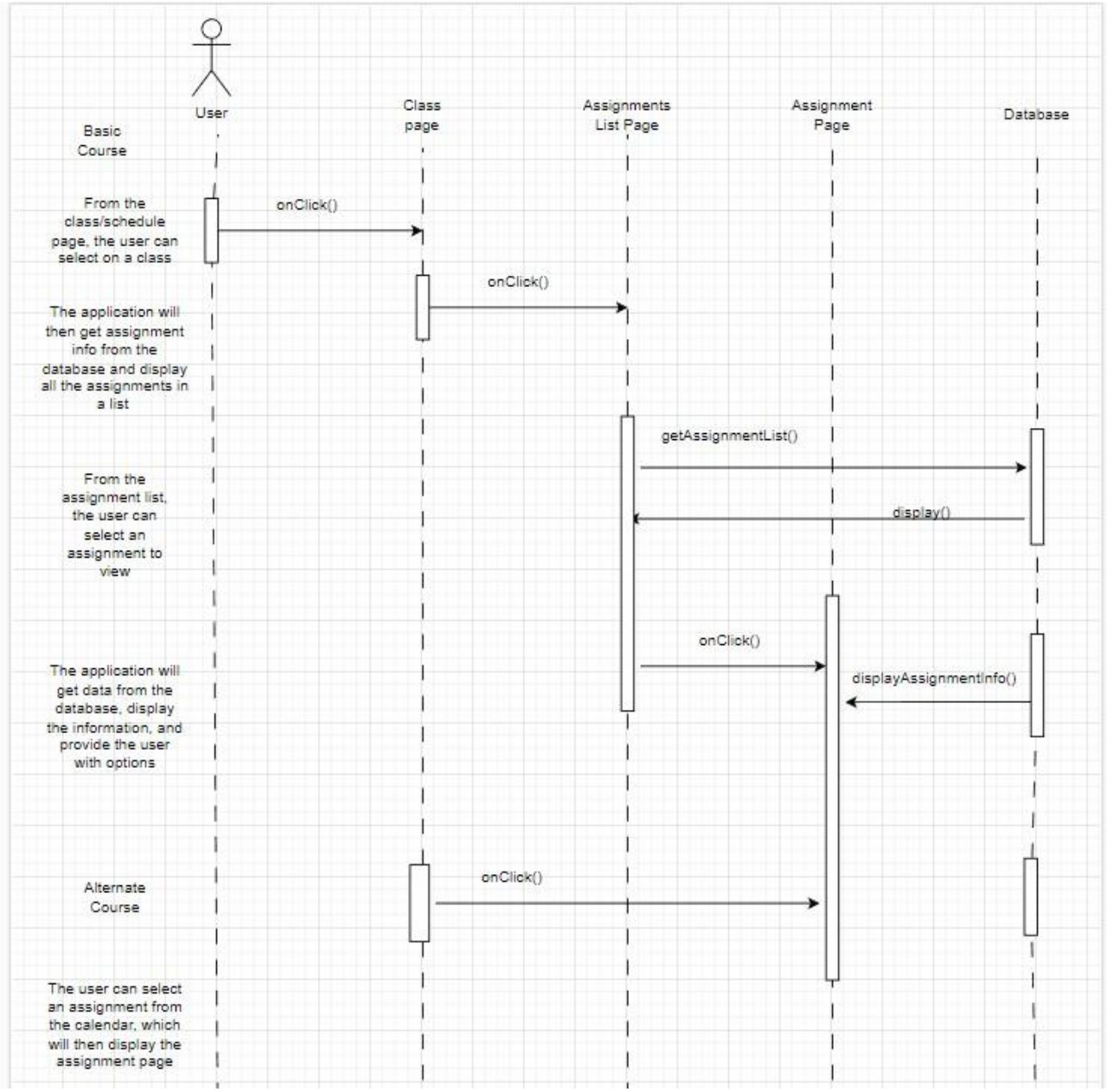
Creating Student/Teacher Account



Assignment Application



Sequence Diagram

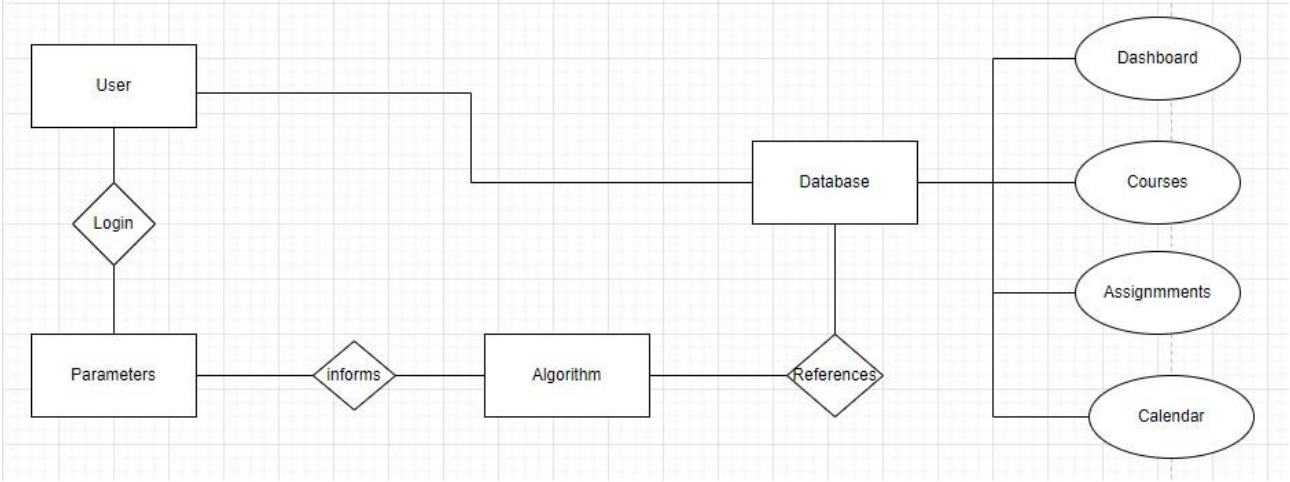


8.3 Design Rationale

We selected that architectural design because it was best fit for our vision of the application. With how our application is being implemented, the uml diagram was how our plan can be accurately visualized. Issues we may encounter are the dependencies between each separate feature may interfere with one another. However we trade with the accuracy and accessibility of each feature and the breakdown in order to create less errors

9. Data Design (only applicable if your project has a database component)

9.1 Entity relationship design

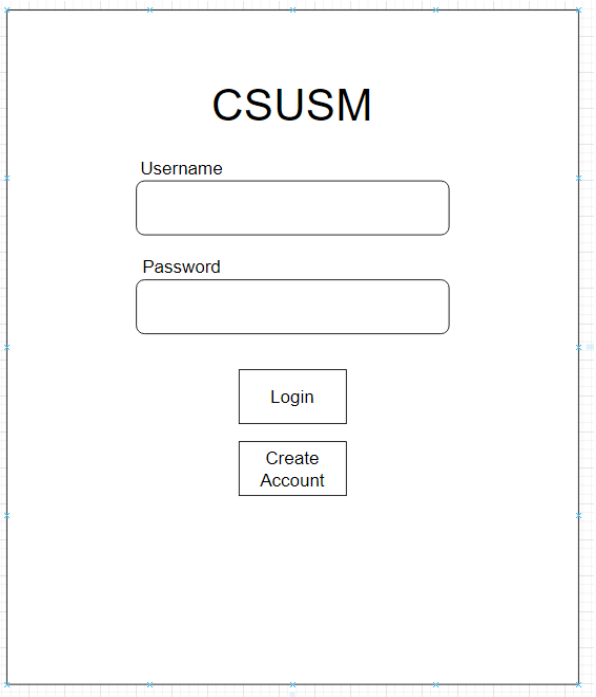


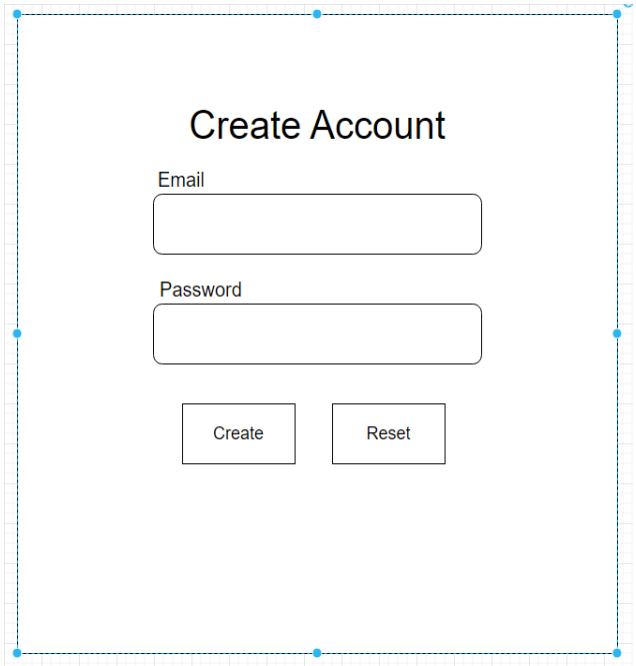
9.2 Data dictionary

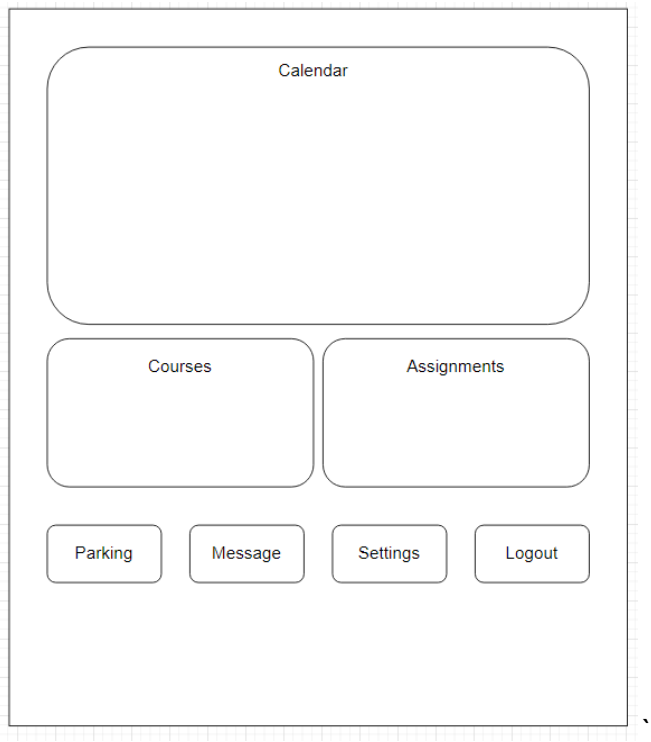
Entity	Attributes	Description
User	userID, firstName, email, role()	store login information of users
Login	userID, password	information required to log in
Inform	messageType	return message for entry
Reference	title, URL	SQL reference material
Database	databaseID, Name, Server	database used by the application
Dashboard	edit, class, assignment	user-specific design and data
Courses	Title, Instructor, startDate, endDate	course information for users
Assignments	assignmentID, courseID, title, dueDate	store class assignments
Calender	startDate, endDate, time()	calender event for users

10. Other Requirements

Appendix A: Screen Images







The first image is our login page with the header “CSUSM” indicating our school. The user will have two available text fields where they will input their username within the “username” textfield and their password associated with their user within the “password” text field. Below the text fields are two buttons. The first button in the login button. Once the user inputs their username and password, they then will click the login button in order to successfully login into their account. However if the requirements aren’t met such as the username and password don’t match then the user will have to retry. The second button is the create account button, this will lead to our create account page.

The second image is our create account page with the header “Create Account”. The user within this page will have two text fields where they will input a valid email address and create a password for their account in order to store it safely within the database. Below the text fields are two buttons. The first button is the create button. Once the user inputs both a valid email and password, clicking the create button will then direct the user to the main dashboard home page. The second button is the reset button. This button will completely reset the create account page clearing the email text field and password text field.

The third image is our dashboard/home page. The top and biggest field will be where the calendar will be displayed. This calendar will inform the user of upcoming events, assignments and other personal information the user may want to include. Below the calendar and to the left is the courses page. The courses page will lead the user to a page where it shows which courses the user is in and able to view to their liking. Below the calendar and to the right is the assignments page. The assignments page will lead the user to a page where it’ll display their upcoming assignments as well the option to view previous assignments. The assignments will be listed from top to bottom with the top having the closest due date to indicate urgency. Below the courses and assignments buttons will be four smaller buttons which will be a parking button to help inform the user about available parking and traffic, message button allowing the user to email teachers or peers, settings button allowing the user to change up their pages to their preferences, and a logout button that will allow the user to safe logout of the application with their information safely stored to their designated account.

These images are all a very rough draft and subject to change as we progress through the development of our application.