Professor and Course Rating Mobile Application

Requirements and Specification Document

2020-02-12, version 1.0

Project Abstract

New and current students would like to choose a course that is related to their interests as well as professors that teach according to their way of studying. However, right now the only way to check is through ratemyprofessors and sometimes it is not accurate because everyone has the right to give comments. The website is also quite simple.

This mobile application will be able to search through all the available courses in UW - Madison, followed by the list of professors that has taught the class before as well as information regarding the class such as the prerequisites and class content. After that, the users of the app would be able to rate the professor by stating their difficulty and performance throughout the class. Other than the rating feature, users will also be able to see how other students rate that professor as one of the considerations on whether they should take the class or not.

Document Revision History

Rev. 1.0 2020-02-12: initial version

Customer

The target user should be all the students of UW Madison including transfer students, new students and current students. The user would also include people that are interested in UW Madison's available courses and professors.

Kevin, a senior here in UW Madison is the informer for this document. He helped us to determine how the application should be designed in terms of how the rating feature should be implemented such as having the difficulty rating as well as the professor rating. We will be talking with Kevin as well as other students throughout the course of the project

Competitive Landscape

There are competitors in the market for example- “Rate my Professors” website. However our application is better than the above mentioned website in a variety of ways:-

- In order to remove toxic as well as spam comments received in the system, “Rate my Professors” has a team of moderators to ensure that such comments are not displayed on the front-end.However, this approach is quite tedious as well as expensive, owing to the salaries paid to the respective employees in order to perform those tasks.While on the other hand our “Professor and course rating” mobile application makes use of natural language processing to filter out the spam and toxic comments from the database, before they are displayed on the front-end.This makes the overall cost of maintaining the application less expensive as well as saves a lot of hassle of filtering out comments manually.
- The “Rate my Professors” website allows any individual, that has an email address to rate a course or a professor on their website, however, our application use special email address authentication in order to ensure that the comment or rating received in the system comes from a legit individual studying in a particular university where the course has been taught.
- The “Rate my Professors” website contains a ton of advertisements and that too at peculiar places on the front-end, which can cause frustration in the mind of the user, however our application only displays advertisements at a particular banner in the application in order to ensure that it does not interfere with normal working of the application and enables the user to perform tasks without getting muddled by the application.

User Requirements

Required Features: These were determined by us and the customers to be the key features needed in the first version of the app

1. User can login and register new account
2. User can search for a specific course
    a. A searching bar in the homepage
    b. List of courses
        i. Course with the same name if found
        ii. Courses with similar names otherwise
3. User can view the introduction of a course and professors who teach the course
    a. Introduction of the course
        i. Prerequisite
        ii. Content of the course
        iii. Requirements the course satisfies
    b. List of all professors who have taught the course
        i. Professor's name
        ii. Average rating of performance in graphic form
        iii. Average rating of difficulty in graphic form
    c. Selecting professors' ratings with courses
    d. Adding the selected professors' ratings to saved list
4. User can check professor's rating detail with a course in course page
    a. Click professor in course page
    b. View average score of performance of the professor in the course
    c. View average score of difficulty of the course with the professor
    d. View comments
    e. Add this professor's rating with the course to saved list
5. Rating professors' performance in a course
    a. Click rating
    b. Select level of difficulty and performance
    c. Leave comment for the professor
6. User can check professor's rating detail with a course in saved list
    a. Click saved list
    b. Click the professor-course pair
    c. View average score of performance of the professor in the course
    d. View average score of difficulty of the course with the professor
    e. View comment
7. User can remove professor's rating from the saved list
    a. Selecting professor's rating in the saved list

    b. Click remove button

2. **Additional Features:** These will only be implemented if the features above can be done. Therefore, we don't have use cases for the features below. Our original version user-flow won't include these features too.

   1. User can view course ranking in the homepage
      a. View all professor-course pairs sorted by professor's performance
      b. View all professor-course pairs sorted by difficulty
   2. User can sort professors for a course
      a. User can select using score of performance or difficulty to sort professors of a course
      b. View sorted professor list in the course page
   3. User can edit their previous rating and comment
      a. Click my rating
      b. View all rating done by the user
      c. Select a rating
      d. Click edit
      e. Rerating

Use Cases

Name	Login and register a new account
Actors	Users who haven't login
Triggers	1. Click on rating button of a professor in course page 2. Click on saving button in course page 3. Click on saved list button from any page
Events	-Click on rating/saving button -Login/register page appears -Login with username and password or register a new account
Exit Condition	a.Click login or Click register b.Click cancel button
Post-conditions	1a.Rating page appears 2a.Notice rating added to the saved list

	3a.Saved-list page appears b.Back to previous page
Acceptance Test	1a.User clicks on the rating button from the course page and a login page appears, after entering the correct username and password or registering a new account, jump back to the rating page. 2a.User clicks on the save button from the course page and a login page appears, after entering the correct username and password or registering a new account, jump back to preview page and a notification of “rating added to the saved list” appears. 3a.User clicks on the saved list button from any page and a login page appears, after entering the correct username and password or registering a new account, jump to the Saved-list page. b.User clicks on the saved list button from any page and a login page appears, click cancel button, jump to previous page.

Name	Searching for a specific course
Actors	User who has logged in or has not logged in
Triggers	Click search button in the home page
Events	-Entering course name in the home page’s searching bar -Click searching button -List of course appears -Scroll through the list and view course(s) name(s)
Exit Condition	a.Click a course b.Click home button
Post-conditions	a.Enter course page b.Back to home page
Acceptance Test	User enters a course name and clicks the search button on the home page, the list of courses matched appears.

Name	View the introduction of a course and professors who teach the course
Actors	User who has logged in or has not logged in
Triggers	Click a course in the list of courses
Events	-Click a course in the list of courses -Course page appears (including course introduction and professor list) -Scroll through the list and view professors who teach the course
Exit Condition	a.Click home button b.Click a professor in the professor list c.Click saved list
Post-conditions	a.Back to home page b.Professor page appears c.Saved-list page appears
Acceptance Test	User clicks a course in the course list, then the course page appears, which includes the course introduction at top, scroll down to see the list of professors.

Name	Check professor’s rating detail from course page
Actors	User who has logged in or has not logged in
Triggers	Click a professor in professor list of a course page
Events	-Click a professor pair -Professor page appears, containing professor’s introduction -Scroll down to see comments and ratings
Exit Condition	a.Click homepage button b.Click return button c.Click saved list button
Post-conditions	a.Homepage appears b.Back to course page c.Saved-list page appears

Acceptance Test	a.User clicks a professor in the professor list of a course page, professor page appears with introduction, comments, and ratings. User clicks the homepage button and the homepage appears. b.User clicks a professor in the professor list of a course page, professor page appears with introduction, comments, and ratings. User clicks the return button and the previous course page appears. c.User clicks a professor in the professor list of a course page, professor page appears with introduction, comments, and ratings. User clicks the saved list button and the Saved-list page appears. If the user has not logged in, the login page will appear.

Name	Selecting ratings of professors and save
Actors	User who has logged in
Triggers	Long press a professor in the professor list of a course
Events	-Long press a professor in the professor list -Selection box and save button appear -Select multiple professor
Exit Condition	a.Click save button b.Click cancel button
Post-conditions	a.Notice items have been added to saved list, stay in course page b.Stay in course page
Acceptance Test	a.User long press a professor in the course page until the selection box appears, select multiple professors and click the save button. b.User long press a professor in the course page until the selection box appears, select multiple professors and click the cancel button.

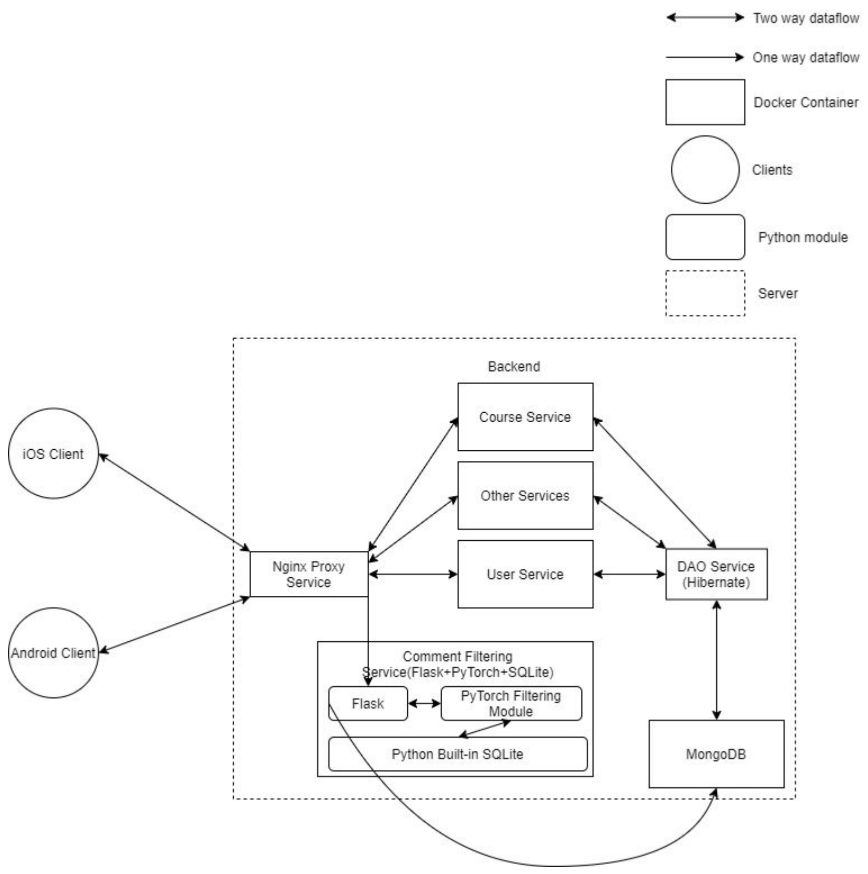
Name	Selecting rating of professors and save

Actors	User who has not logged in
Triggers	Long press a professor in the professor list of a course
Events	-Long press a professor in the professor list -Selection box, save, and cancel button appear -Select multiple professor
Exit Condition	a.Click save button b.Click cancel button
Post-conditions	a.Jump to “Login and register a new account” use case. Notice items have been added to the saved list if logged in. b.Stay in course page
Acceptance Test	a.User long press a professor in the course page until the selection box appears, select multiple professors and click the save button. Then login or register a new account, and jump back to the course page. b.User long press a professor in the course page until the selection box appears, select multiple professors and click the cancel button. Stay at the course page.

Name	Rating professors’ performance in a course
Actors	User who has logged in
Triggers	1.Click rating button of a professor in a course page’s professor list 2.Click rating button of a professor in a professor page
Events	-Click rating button -Rating window appears -Select level of difficulty and performance -Write comment
Exit Condition	a.Click submit button b.Click cancel button
Post-conditions	a.Notice rated, stay in course page b.Stay in course page
Acceptance Test	1a.User clicks the rating button of a professor in the professor list of the course page. Then select the level of

	difficulty and performance in rating window, write comment in rating window. Click submit, notice appears and the course page is updated. 1b.User clicks the rating button of a professor in the professor list of the course page. Then select the level of difficulty and performance in rating window, write comment in rating window. Click cancel, the course page remains. 2a.User clicks the rating button in a professor page. Then select the level of difficulty and performance in rating window, write comment in rating window. Click submit, notice appears and the professor page updated. 2b.User clicks the rating button in a professor page. Then select the level of difficulty and performance in rating window, write comment in rating window. Click cancel, the professor page remains.

Name	Rating professors’ performance in a course
Actors	User who has not logged in
Triggers	1.Click rating button of a professor in a course page’s professor list 2.Click rating button of a professor in a professor page
Events	-Click rating button -Jump to login page -Rating window appears -Select level of difficulty and performance -Write comment
Exit Condition	a.Click submit button b.Click cancel button c.Canceled in login page
Post-conditions	a.Notice rated, stay in course page b.Stay in course page c.Stay in course page
Acceptance Test	1a.User clicks the rating button of a professor in the professor list of the course page. Login page appears, user login or create a new account. Then select the level of difficulty and performance in rating window, write comment in rating window. Click submit, notice appears and the

	course page is updated. 1b.User clicks the rating button of a professor in the professor list of the course page. Login page appears, user login or create a new account. Then select the level of  Two way dataflow One way dataflow Docker Container Clients Python module Server Backend Course Service Other Services User Service DAO Service (Hibernate) iOS Client Android Client Nginx Proxy Service Comment Filtering Service(Flask+PyTorch+SQLite) Flask PyTorch Filtering Module Python Built-in SQLite MongoDB difficulty and performance in rating window, write comment in rating window. Click cancel, the course page remains. 1c.User clicks the rating button of a professor in the professor list of the course page. Login page appears, user click cancel. The course page remains. 2a.User clicks the rating button in a professor page. Login page appears, user login or create a new account. Then select the level of difficulty and performance in rating window, write comment in rating window. Click submit, notice appears and the course page is updated. 2b.User clicks the rating button in a professor page. Login page appears, user login or create a new account. Then select the level of difficulty and performance in rating window, write comment in rating window. Click cancel, the course page remains. 2c.User clicks the rating button in a professor page. Login page appears, user click cancel. The course page remains.

Name	View professor-course pairs added to the saved list
Actors	User who has logged in
Triggers	Click saved list button
Events	-Click saved list button -Saved-list page appears containing list of professor-course pairs
Exit Condition	a.Click one professor-course pair b.Click home page button
Post-conditions	a.Enter professor page, rating and comment of the professor’s performance on the specific course listed b.Home page appears
Acceptance Test	a.User clicks saved list, Saved-list page appears, user clicks a professor-course pair, jumps to professor page. b. User clicks saved list, Saved-list page appears, user clicks homepage button, jumps to homepage.

Name	View professor-course pairs added to the saved list
Actors	User who has not logged in
Triggers	Click saved list button
Events	-Click saved list button -Jump to login page -Saved-list page appears containing list of professor-course pairs
Exit Condition	a.Click one professor-course pair b.Click home page button c.Canceled in login page
Post-conditions	a.Enter professor page, rating and comment of the professor’s performance on the specific course listed b.Home page appears c.Stay at the original page

Acceptance Test	a.User clicks saved list, the login page appears, user login or create a new account. Saved-list page appears, user clicks a professor-course pair, jumps to professor page. b.User clicks saved list, the login page appears, user login or create a new account. Saved-list page appears, user clicks homepage button, jumps to homepage. c.User clicks saved list, the login page appears, user click cancel. The previous page appears.

Name	Check professor’s rating detail with a course in the saved list
Actors	User who has logged in
Triggers	Click a professor-course pair in the saved list
Events	-Click a professor-course pair -Professor page appears, containing professor’s introduction -Scroll down to see comments and ratings
Exit Condition	a.Click homepage button b.Click return button
Post-conditions	a.Homepage appears b.Saved-list page appears
Acceptance Test	a.User clicks a professor-course pair in the saved list, professor page appears with introduction, comments, and ratings. User clicks the homepage button and the homepage appears. b.User clicks a professor-course pair in the saved list, professor page appears with introduction, comments, and ratings. User clicks the saved list button and the saved page appears.

Name	Remove professor’s rating from the saved list
Actors	User who has logged in
Triggers	Long press a professor’s rating in the saved list
Events	-Long press a professor-course pair in the saved list

	-Selection box, remove, and cancel button appear -Select multiple professor-course pairs
Exit Condition	a.Click remove button b.Click cancel button
Post-conditions	a.Saved-list page updated b.Saved-list page remain
Acceptance Test	a.User long press a professor-course pair in the Saved-list page, select multiple professor-course pairs and click remove button. Saved-list page updated and the selected professor-course pairs disappeared. b.User long press a professor-course pair in the Saved-list page, select multiple professor-course pairs and click cancel button. Selection box, remove button, and cancel button disappear, Saved-list page stay as before.

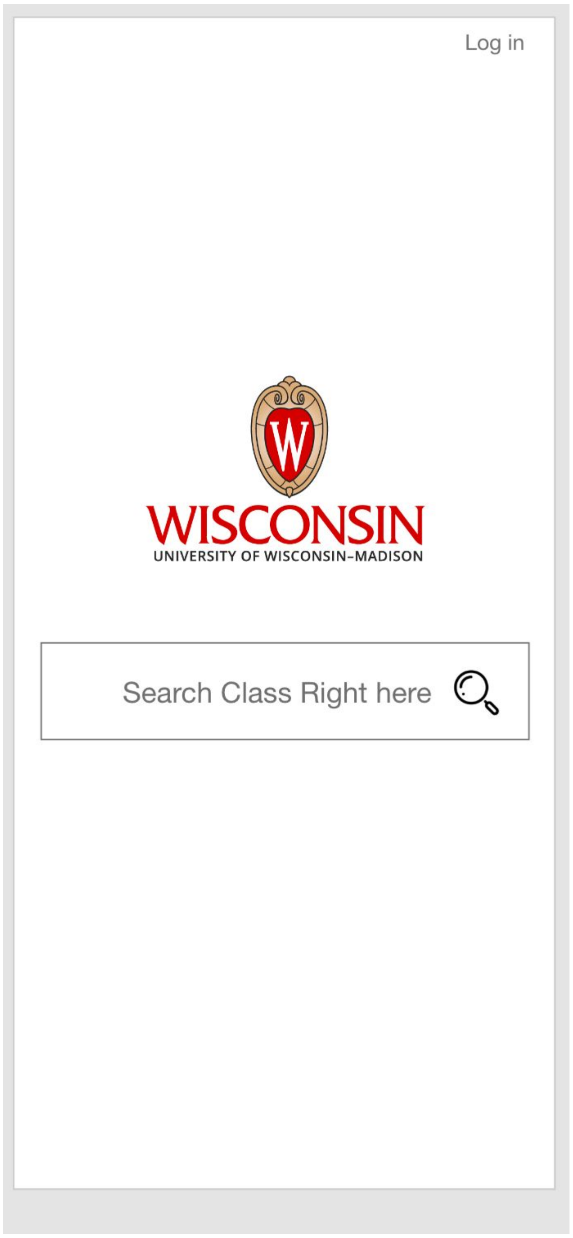
User Interface Requirements

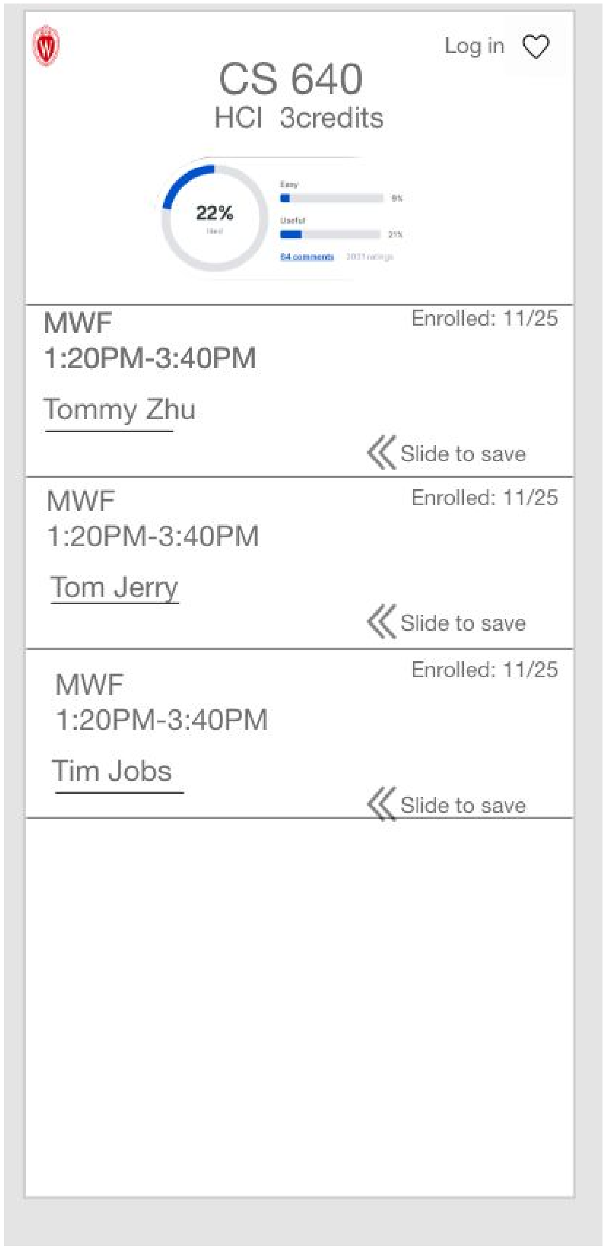
For each page, there should be a button to let users go to the homepage and saved-list page.

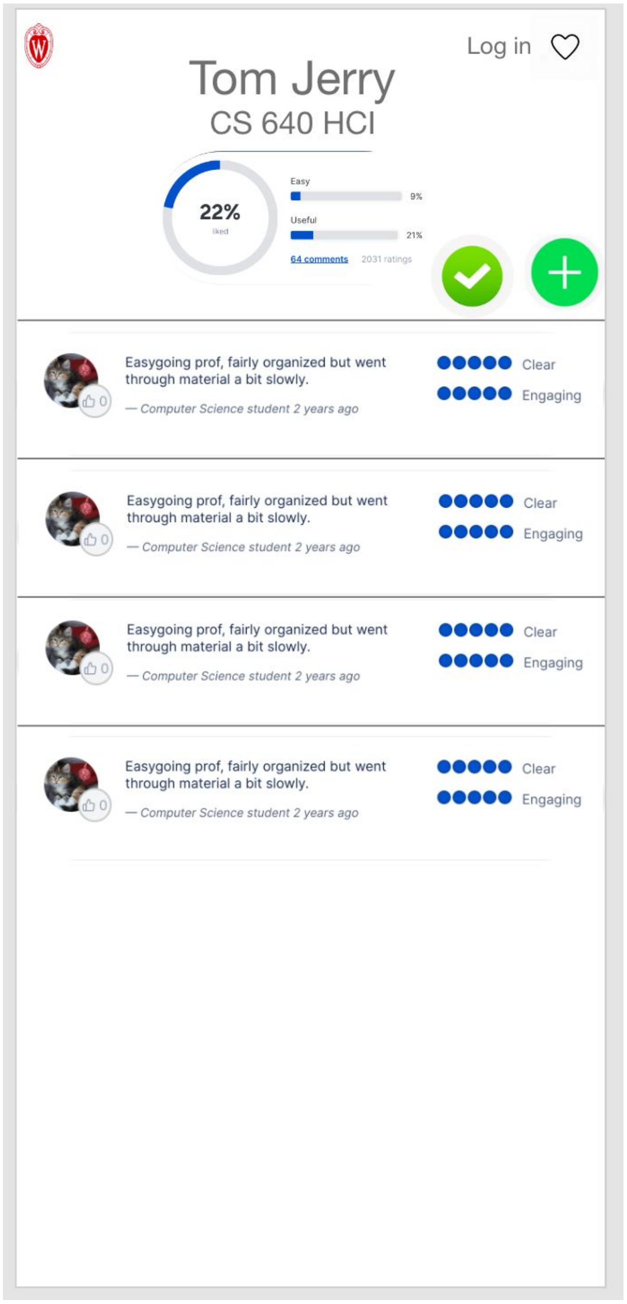
In the homepage, the user should be able to input the course name into a search bar.

In the course page, the user should be able to see all the professors that teach the course. (If the user doesn't log in, he can only review three options.) The user can check the course basic information on this page. Once the user clicks the specific pair of professors and courses, the user can check comments for this course and professor. (If the user doesn't log in, he can only review three reviews.)User can click the add button to leave a comment. (Users who don't log in cannot leave a comment.) The user can click the green click button to save this course. (Users who don't log in cannot leave a comment.) The input is the course name from the search bar of the homepage and the output are the matched courses.

Users can click the save list icon on the top of the right to go to the saved list page. In the saved list page, the user should be able to see all the courses that they selected before and with the course and professors rating. (Users who don't log in cannot go to this page.)


Log in
WISCONSIN
UNIVERSITY OF WISCONSIN–MADISON
Search Class Right here


Log in
CS 640
HCI 3credits
22%
liked
Easy
9%
Useful
21%
84 comments
2031 ratings
MWF
1:20PM-3:40PM
Enrolled: 11/25
Tommy Zhu
Slide to save
MWF
1:20PM-3:40PM
Enrolled: 11/25
Tom Jerry
Slide to save
MWF
1:20PM-3:40PM
Enrolled: 11/25
Tim Jobs
Slide to save



Log in

Tom Jerry

CS 640 HCI

22%
liked

Easy 9%

Useful 21%

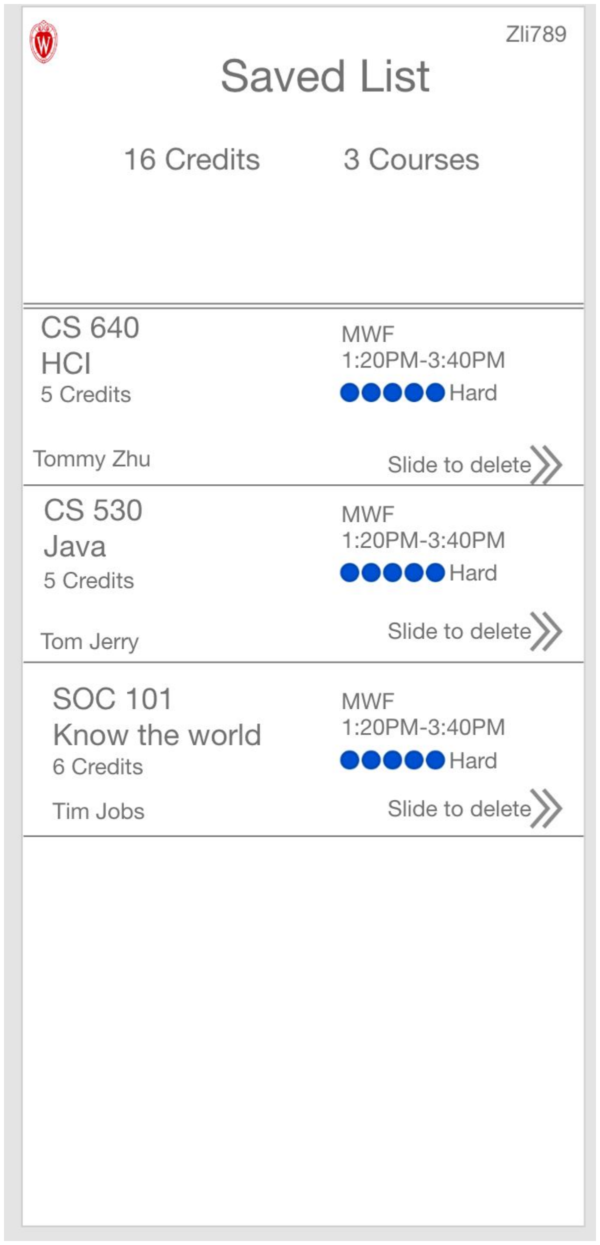
64 comments 2031 ratings

Easygoing prof, fairly organized but went through material a bit slowly.
— Computer Science student 2 years ago
●●●●● Clear
●●●●● Engaging

Easygoing prof, fairly organized but went through material a bit slowly.
— Computer Science student 2 years ago
●●●●● Clear
●●●●● Engaging

Easygoing prof, fairly organized but went through material a bit slowly.
— Computer Science student 2 years ago
●●●●● Clear
●●●●● Engaging

Easygoing prof, fairly organized but went through material a bit slowly.
— Computer Science student 2 years ago
●●●●● Clear
●●●●● Engaging



Zli789

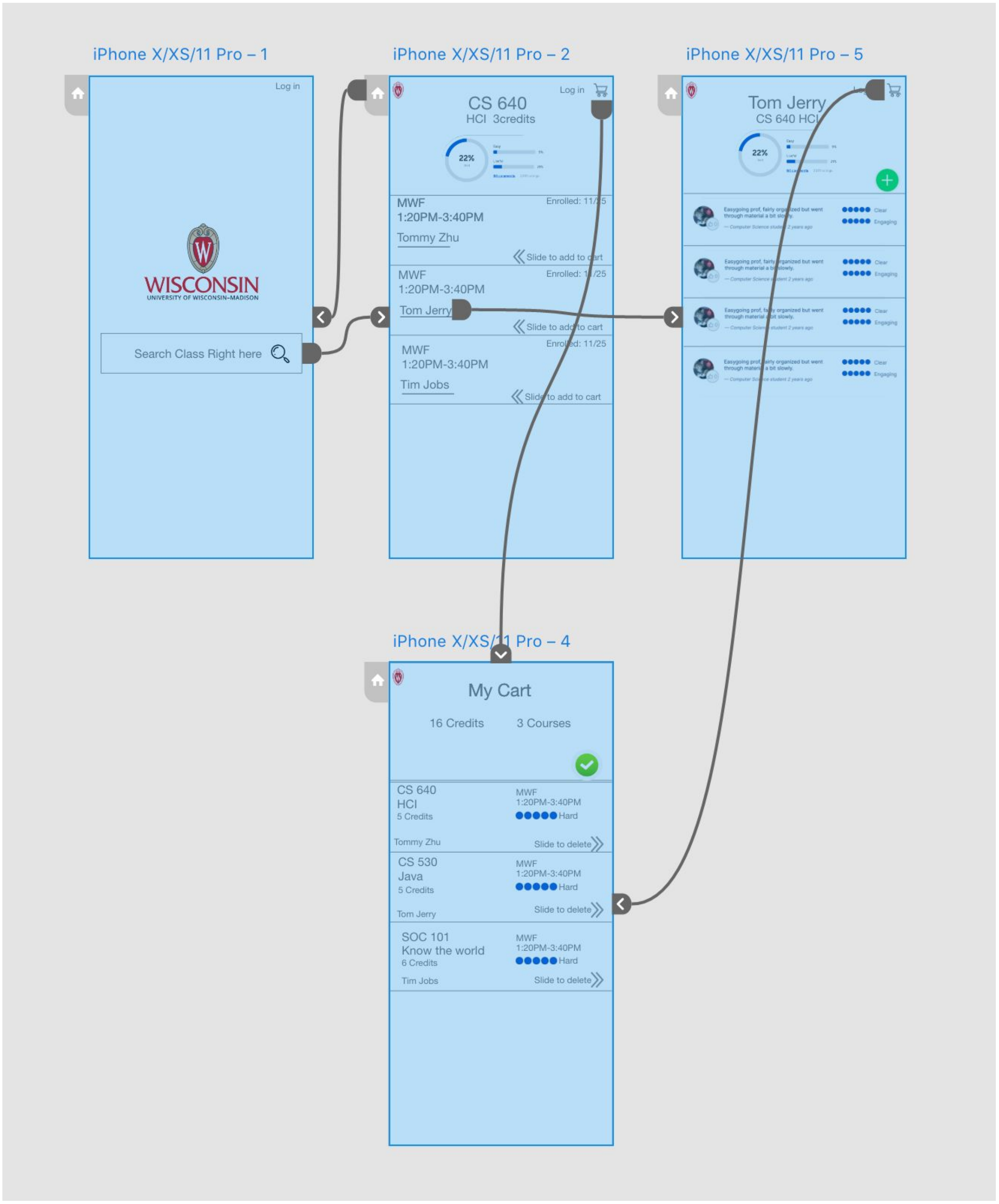
Saved List

16 Credits 3 Courses

CS 640
HCI
5 Credits
MWF
1:20PM-3:40PM
●●●●● Hard
Tommy Zhu
Slide to delete

CS 530
Java
5 Credits
MWF
1:20PM-3:40PM
●●●●● Hard
Tom Jerry
Slide to delete

SOC 101
Know the world
6 Credits
MWF
1:20PM-3:40PM
●●●●● Hard
Tim Jobs
Slide to delete

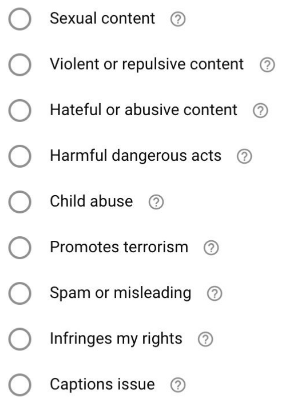

iPhone X/XS/11 Pro – 1
Log in
WISCONSIN
UNIVERSITY OF WISCONSIN–MADISON
Search Class Right here
iPhone X/XS/11 Pro – 2
Log in
CS 640
HCI 3credits
22%
MWF
1:20PM-3:40PM
Enrolled: 11/25
Tommy Zhu
Slide to add to cart
MWF
1:20PM-3:40PM
Enrolled: 11/25
Tom Jerry
Slide to add to cart
MWF
1:20PM-3:40PM
Enrolled: 11/25
Tim Jobs
Slide to add to cart
iPhone X/XS/11 Pro – 5
Tom Jerry
CS 640 HCI
22%
Easy
Easygoing prof, fairly organized but went through material a bit slowly.
— Computer Science student 2 years ago
Clear
Engaging
Easygoing prof, fairly organized but went through material a bit slowly.
— Computer Science student 2 years ago
Clear
Engaging
Easygoing prof, fairly organized but went through material a bit slowly.
— Computer Science student 2 years ago
Clear
Engaging
Easygoing prof, fairly organized but went through material a bit slowly.
— Computer Science student 2 years ago
Clear
Engaging
iPhone X/XS/11 Pro – 4
My Cart
16 Credits
3 Courses
CS 640
HCI
5 Credits
MWF
1:20PM-3:40PM
Hard
Tommy Zhu
Slide to delete
CS 530
Java
5 Credits
MWF
1:20PM-3:40PM
Hard
Tom Jerry
Slide to delete
SOC 101
Know the world
6 Credits
MWF
1:20PM-3:40PM
Hard
Tim Jobs
Slide to delete

Security Requirements

As the application involves the data of all the professors, teaching assistants, hence it is necessary to ensure that only the teachers and students (incoming and current) can fully use it. Only teachers and students can view and leave comments on courses as well as professors. Otherwise, people can only check course information and a part of the professors' information (show 3 comments).

Therefore, in order to view the information on the application, the user needs to use his/her respective Wisconsin email-id for login purposes, without which they will not be able to access all the features of the application.

In order to avert detrimental comments from being displayed on the front-end, a machine learning model will be used to filter all the comments before they are displayed on the front-end. Detrimental comments include any context with the following list.



- Sexual content
- Violent or repulsive content
- Hateful or abusive content
- Harmful dangerous acts
- Child abuse
- Promotes terrorism
- Spam or misleading
- Infringes my rights
- Captions issue

There are also a ton of spam messages out there that can disrupt the motive behind an application, hence in order to stop spam messages from accumulating in the application, a spam filtering machine learning model will be used to filter the spam messages out of the application. The machine learning team will think of a rule to make sure that any detrimental comments won't show up on the front-end, and it won't affect that student leaving a negative comment for professors. Spam messages will be 100% filtered out by the model.

In order to improve the scalability, the back-end consists of multiple microservices, each independent of each other. Therefore, even if one service goes down, other service can run independently. This could effectively prevent malicious attack on the server.

System Requirements

The system will include two parts: client-side and server side.

Client(Front-end):

Since the target client will be running on both iOS and Android systems. According to the first specification meeting, the front-end will be implemented in either **ReactNative**, which uses **JSX**.

Considering the overall resource cost of the user interface, the minimum requirement for an Android device is Android 4.0 with Internet access, and for iOS is iPhone 5. For both Android and iOS, the minimum requirement for storage is ~50mb.

Server(Back-end):

The server side will be deployed on possibly multiple **Docker** containers. The docker engine will be running upon a Linux(**Ubuntu 18.04**) server. DigitalOcean or Vultr VPS may be considered for development and demonstration, using my own server at the apartment otherwise.

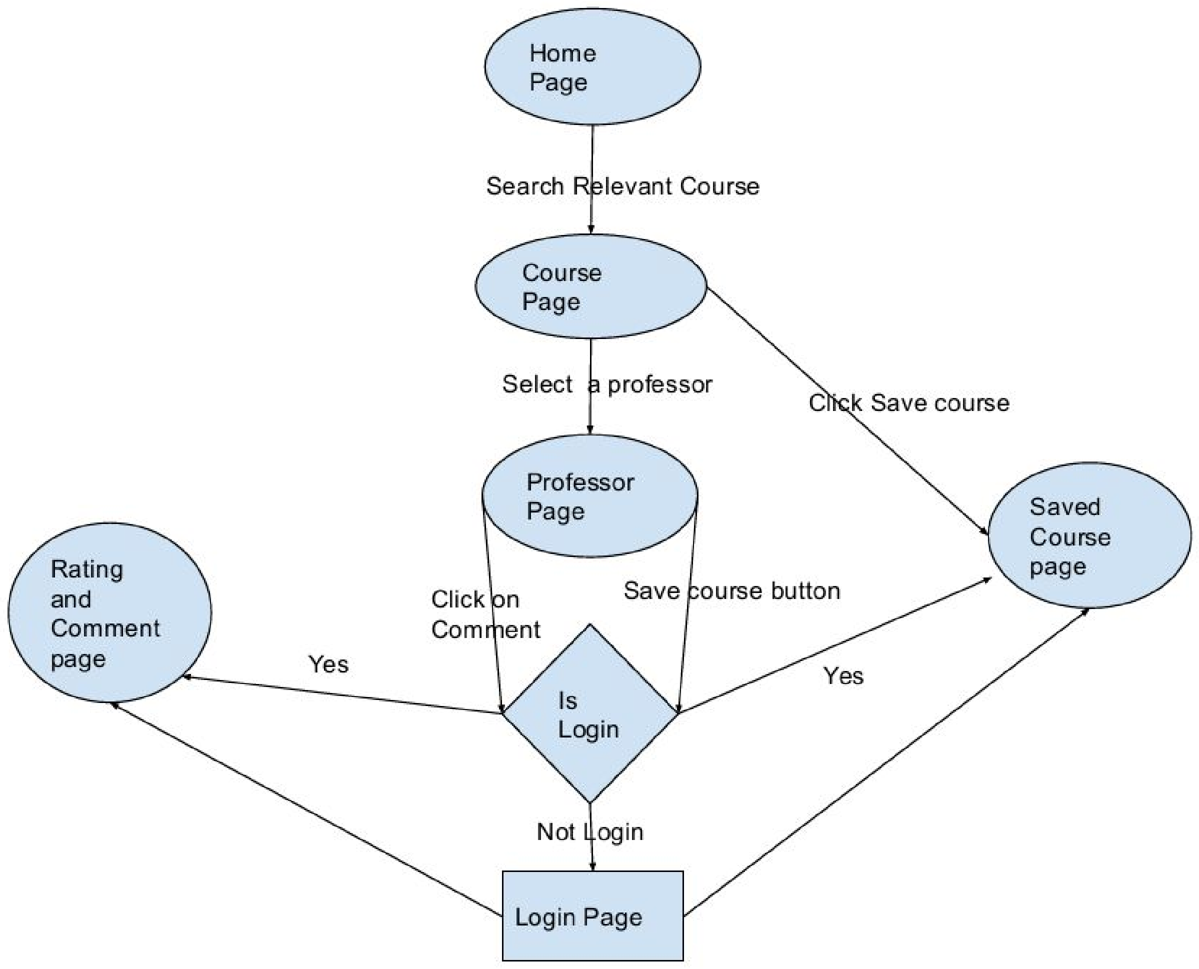
Tech stacks for the server-side:

- Server
  - Ubuntu 18.4(Either use VPS or Daniel’s server at his apartment)
    - **Docker Engine** on Ubuntu 18.4
      - **Nginx** Container for redirecting requests
      - Microservices built on **Spring Boot**(Java)
        - Specifically, for UserService, **JWT** will be used for authentication
      - **Hibernate** for database access
      - **MongoDB**
      - Filtering Service in **Python**
        - **Flask** for communicating with other services
        - **Keras** for machine learning kit

    ○ **SQLite**, the built-in SQLite for machine learning data

*All Docker Containers run on the local Docker Engine, which will be running on the only Ubuntu server.

Specification


Home Page
Search Relevant Course
Course Page
Select a professor
Click Save course
Professor Page
Saved Course page
Rating and Comment page
Click on Comment
Save course button
Yes
Is Login
Yes
Not Login
Login Page