**CSCB07 Group 6 Master Document**

Shanni Li, Xinyu Liu, Juno Zhang, Ruofan Chen, Rebecca Han

(Workload percentage: 20% each among all group members)

**Main tasks done by each member of the group**

* Ruofan Chen (scrum master)
  + Designed & Implemented Login & Sign In parts of the app, including the initial connection to firebase (code later used as template by other members), all UI sections of the code (LoginActivity, LoginFragment, SignUpFragment), all java code for login & sign in, and navigation between login & sign up
  + Refactored these code based on MVP principles to follow the instructions on testing (User, Login Model, Login Presenter, Login Fragment, etc.)
  + Fully tested the Login code by writing the JUnit tests
  + Helped group members with debugging (mainly with admin firebase related issues)
  + Helped with merging the different branches & sections of the app (navigations between login to the student/admin)
  + Updated and modified SCRUM related tasks (e.g., JIRA, agreement, notetaking during meetings)
* Xinyu Liu
  + Discussed and coordinated with Shanni to make sure the input types we used on Student side are the same, and made agreements on the format of displayed information.
  + Designed and implemented Java class: CourseList which is used to store information of each course, FutureCourse which is used to store courses that the student wants to take, and PastCourses which is used for storing courses already taken before.
  + Implemented java code for adding , deleting and viewing courses for Student part.
  + Implemented generatingTimeline method for student part by dividing it into first generating course list and then arranging courses into respective semesters, and creating Java Class Timeline as the return type of generatingTimeline.
  + Tested addCoursed, deleteCourses, viewCourses, and generatingTimeline using various test cases in Eclipse.
  + Tested generating timeline in Android Studio using firebase, and solved bugs.
* Shanni Li
  + Designed and completed all UI and framework including buttons, textview, input spaces and navigation between activities of all student parts on Android Studio.
  + Implemented and completed functions including viewing, adding, deleting past and future courses on Android Studio.
  + Created a google realtime firebase, setted it to run successfully, and linked it with the student part project on Android Studio.
  + Imported the data that the student part needed from the firebase, and transformed it to the format the student part needed. Ensured the data can be updated in the app when the corresponding data on firebase was updated. Solved some necessary problems to ensure the data from firebase can be used in other places.
  + Fully tested on viewing, adding, deleting past and future courses and generating timeline, and debugged on Android Studio, making the app works as I expected.
  + Add sufficient restrictions and conditions on all students’ input to ensure all inputs are valid and avoid the app terminating accidentally for all functions.
  + Cleaned code and generated the final version.
* Rebecca Han
  + Designed the Admin functionalities alongside Juno Zhang
  + Implemented and debugged the AdminCreate class as well as its corresponding layout file. Wrote and tested functions regarding the creation of a course based on user input and storing it into Firebase. Validated the data: no duplicate courses, no prerequisites of a higher level may be added (e.g. "CSCA48" should not be a prerequisite for "CSCA08"), and no blank courses. This also ensures that the Student functions work as desired
  + Implemented and debugged the AdminEdit class as well as its corresponding layout file. Wrote and tested functions regarding the fetching of an existing course and then storing user-made edits to it into Firebase
  + Handled the fragment navigation between Admin-side fragments
  + Optimized UI for clean and easy-to-navigate app use alongside Juno Zhang.
* Juno Zhang
  + Merged all the branches and fixed merge conflicts on Github.
  + Discussed overall class structure with team, created class structure diagram and explicitly programmed the Course class
  + Implemented Admin functionality with AdminViewCourses, AdminViewFragment, and AdminViewModel where the user can view the current courses stored in the database as a list of buttons
    - Within this display, each button can be double-clicked to remove the course from the database
  + First tested this value with a dummy set of values, before connecting it to the firebase database
  + Used debug mode to figure out issues related with firebase loading values. Turns out, because of multi-threading, you can’t read values from a database and then refer to it in code immediately after, because the values won’t be there yet :/
  + Polished the final product by adding some colours and design

**Scrum Meeting Summaries**

Sprint meeting (November 16)

* 2 sprints for the remainder?
  + Until Nov 27th
    - Implement basic functionalities of the software
      * Login: firebase setup; login & sign up
      * Admin: create 1+ activity where admin can add/edit/remove courses
      * Student: view current list courses taken & generation of timeline
  + Until Dec 5
    - Done together during the zoom session (merging)?
    - Firebase adding to admin & student
    - Connection of the activities & hooking up to firebase
* Standardized definition of course
* Both working together for user stories
* Determined firebase storing

November 23rd, 2022 (Scrum Meeting, not sprint meeting)

(Student: Shanni, Xinyu) still discussed the data structure and all function we may need and may need to be implemented

(Admin: Rebecca, Juno) Began coding the fragment UIs, and created the Course object with a blank constructor, a filled constructor (code, name, bool[] of offering sessions, List<Course> of prerequisites), and a static list of Course objects to represent the course list.

(Login: Ruofan) I was sick and missed the group tutorial meeting; but I catched up with the discussed information after.

Sprint Meeting 2 (November 27th, 2022)

Things completed:

* Ruofan finished log-in portion, firebase username password storage + junit testing for navigation, also merged student branch into login2
* Admin finished all core functionality, going to integrate firebase
* Students finished all core functionality, haven’t pushed student timeline functionality yet but finished, going to integrate firebase
* finished stories 6, 2 3

To-do:

* synchronize login2 branch into main (once program is completely finished)
* beautiffy UI on Sunday, or whenever all code is finished (Discuss style, juno will implement style)

Discussing contribution of work

* 20% each (even split)

**SCRUM Team Agreement**

* Group Members (Last Name Alphabetical)
  + Ruofan Chen (Scrum Master), Rebecca Han, Shanni Li, Xinyu Liu, Juno Zhang
* Communication
  + Discord (text chat)
  + Zoom (video chat)
  + Cell (emergency)
  + Google Doc (document sharing)
  + GitHub Repository (project sharing/modification/version control)
* Daily Stand-Up
  + Via discord chat & zoom conferences
  + Logged by Scrum Master or self-recorded & updated to the GitHub repository
* Sprints
  + Nov 16-27, and last week Nov 27-Dec 5
  + 4 & 2 user stories should be taken & fulfilled per sprint respectively
  + It is possible that the workload for each member looks different for each sprint; if this is the case, please communicate if you would like someone to help
* Typical Working Hours/Commitment
  + Due to different schedules, other than the required daily stand-up and weekly sprints, the working hours are flexible
  + Group members should try to work such that we obtain a good mark in the project
* User Stories
  + 1 As a **student**, I want to sign up and login to my account so that I can securely access my data. (Ruofan)
  + 2 As a **student**, I want to view the list of courses I have taken and add to this list so that I can keep track of all the courses I have taken. (Shanni & Xinyu)
  + 3 As a **student**, I want to generate a course timeline by providing courses I want to take so that I can plan my education accordingly. (Shanni & Xinyu)
  + 4 As an **admin**, I want to login to my account so that I can securely manage course information. (Ruofan)
  + 5 As an **admin**, I want to add a course and define its name, course code, offering sessions, and prerequisites so that a student’s timeline could be generated correctly. (Rebecca & Juno)
  + 6 As an **admin**, I want to view the list of all courses and edit or delete any course in the list so that I can keep the course information up to date. (Rebecca & Juno)
* Other rules
  + Please **focus on the work** when we are working; the earlier we can finish this, the more time we have later to modify/test it (or just relax!)
  + Opinions of all group members should be considered; however, please differentiate between **suggestions** (problem-based & with a tangible solution) and **criticism** (often people based & without a tangible solution)
    - If you say “no” to a proposed item, you’re expected to try to make it better
  + This is an open & respectful space; we will try to help each other out despite the fact that marks are based on individual contribution
    - If you are not catching up with something, feel free to tell the group about the situation; we will work something out!
    - If there are conflicts of ideas, please assume positive intent & try to resolve the issue professionally & calmly
  + When making group decisions, we can vote on the ideas/proposals
  + If there are problems with the project specifications, please ask & verify! It’s no biggie!
    - Questions regarding the project details can be asked through **Piazza (more general questions)**, or can be assembled to be asked during **in-person office hours (Mondays & Fridays)**
      * I (Ruofan) can ask it for you if you can’t make it time-wise, but you have to explain your problem to me if I cannot understand what it is you are asking

**Daily Stand-up Logs**

November 16th, 2022

As a group, we met together, chose a scrum master, and discussed the rough idea of team agreements, including splitting functionalities/user stories (admin (Juno, Rebecca), student (Shanni, Xinyu), and login (Ruofan)) responsibilities temporarily. Exchanged ideas regarding the program design (mainly UI aspects of functionalities).

November 17th, 2022

(Admin: Juno, Rebecca) Compiled a Google document with a complete list of tasks to do based on the given user stories, and began discussing possible pseudocode that would preserve SOLID principles.

(Student: Shanni, Xinyu) We discussed the design patterns to be used and the general algorithm of the functionality to be implemented according to the user stories, as well as the data edge cases to be concerned about.

(Login: Ruofan) Wrote the agreement for team & thought of basic design of UI

November 18th 2022

(Admin: Juno, Rebecca) Went on call to discuss the general structure of the Admin functionality and the division of labour.

(Student: Shanni, Xinyu) Discussed the data structure and all information the student may need when writing functions.

(Login: Ruofan) Designed the user data structure.

November 19th, 2022

(Admin: Juno, Rebecca) Continued to discuss the general structure of the Admin functionality and the division of labour.

(Student: Shanni, Xinyu) Continued discussing the information we need and user stories to ensure we will not miss required information in future

(Login: Ruofan) Started with the basic UI design of the fragments (by creating the fragments in Android studio) & creation of user data structure class.

November 20th, 2022

(Admin: Juno, Rebecca) Went on call to discuss the general structure of the Admin functionality and the division of labour.

Student

Shanni: Designed the basic UI in student part including view past course and view the courses students want to take and created a real time firebase

Xinyu: Start to thinking about generating timeline

(Login: Ruofan) Finished with the basic UI design of the fragments (by creating the fragments in Android studio) & implementing more basic user structure helper functions.

November 21st, 2022

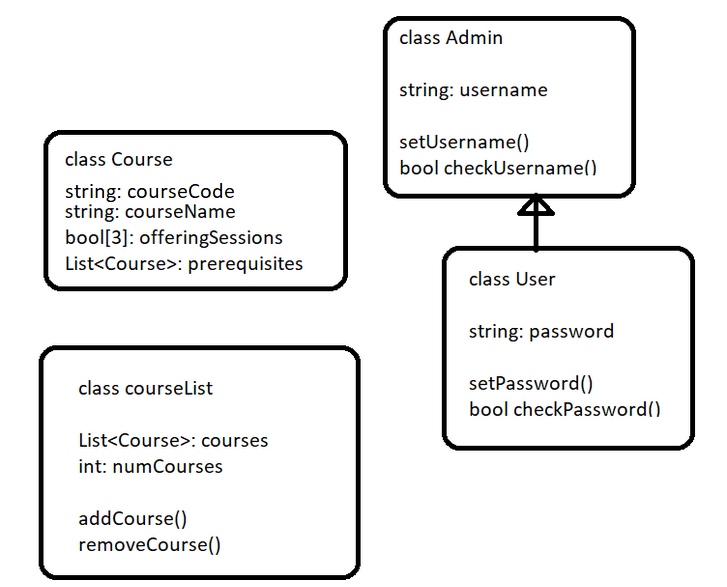
(Login: Ruofan) Started to test with firebase; encountered a lot of issues.

Shanni: optimized the basic UI in student part including view past course and view the courses students

Xinyu: Continued to think about generating timeline

Juno: compiled the following class diagram

<https://docs.google.com/document/d/1aSvuRu0DPYFvvlOm2ICWAF13jmNgPeuMV5zyVFjtAA8/edit> - Rebecca (with input and discussion with Juno)



November 22nd, 2022

(Student: Shanni, Xinyu) discussed the data structure and all function we may need and may need to be implemented

(Login: Ruofan) Still encountered issues with getting & setting data on firebase realtime database.

November 23rd, 2022 (Scrum Meeting)

(Student: Shanni, Xinyu) still discussed the data structure and all function we may need and may need to be implemented

(Admin: Rebecca, Juno) Began coding the fragment UIs, and created the Course object with a blank constructor, a filled constructor (code, name, bool[] of offering sessions, List<Course> of prerequisites), and a static list of Course objects to represent the course list.

(Login: Ruofan) I was sick and missed the group tutorial meeting; but I catched up with the discussed information after.

November 24th, 2022

(Login: Ruofan) Still trying to fix firebase.

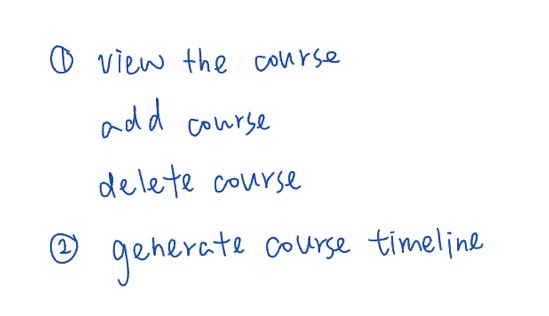
Admin:

Rebecca - Coded navigation between different admin fragments, created skeleton UI for adding a new course, editing an existing course

Juno - Created and filled a testing suite for the admin functionalities, created the AdminFragment that contains/leads to all admin-related code.

(Student: Shanni, Xinyu)

Finally break down into multiple methods



November 25th, 2022

(Login: Ruofan) Started implementing the actual behaviour after buttons being pressed after finally having a working firebase database.

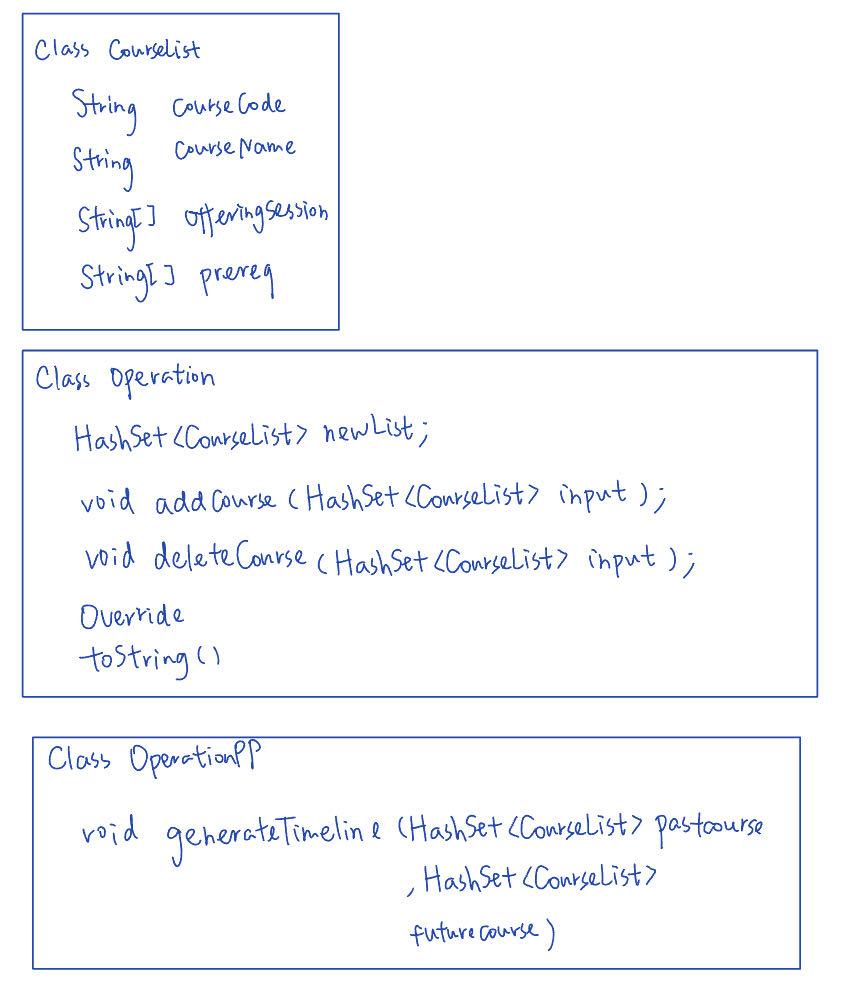
Admin:

Juno - Merged Rebecca’s navigation admin code into the Admin branch

Rebecca - Worked on adding a new course from admin side.

(Student: Shanni, Xinyu)

Wrote some code and function head, finally decided to use Hashset to be the data type of course list:



Sprint Meeting 2 (November 27th, 2022)

Things completed:

* Ruofan finished log-in portion, firebase username password storage + junit testing for navigation, also merged student branch into login2
* Admin finished all core functionality, going to integrate firebase
* Students finished all core functionality, haven’t pushed student timeline functionality yet but finished, going to integrate firebase
* finished stories 6, 2 3

To-do:

* synchronize login2 branch into main (once program is completely finished)
* beautiffy UI on Sunday, or whenever all code is finished (Discuss style, juno will implement style)

Discussing contribution of work

* 20% each (even split)

November 28th, 2022

Student

Shanni: start to implement the function with UI in Android Studio and navigation of buttons and setting text view

Xinyu: Implemented the helper methods for the first part of generating timeline method: findObj, hasPre, seperateByLevel,addToList.

Rebecca: Continued working on adding a new course and implemented storing data into a temporary static arraylist (for testing purposes to be used until firebase is fully implemented)

(Login: Ruofan) Finished login functionalities within the fragments & tested it with some data

November 29th, 2022

Juno - created backend logic (a function) to render all the courses in a given list on the screen as interactive buttons. Created an adding function, that will add a course to a given course list, as well as re-render the buttons to reflect changes. Created a removing function that works similarly.

\*known issues\* buttons can only display up to a max of 15. gonna try to use multiple linearlayout instances to make a sort of grid

Rebecca - Debugged AdminCreate (adding new course), polished some functions

Shanni - Completed viewing, adding and deleting past courses and future courses as students on Android Studio with all features including appropriate navigation activities. Linked the student branch to a realtime firebase.

Xinyu - implemented CourseList class, implemented add, delete, view courses java code, and tested on it. Also completed the code for the first part of generating the timeline - finding the order of courses that students should take, and tested the code.

(Login: Ruofan) Refactored login into MVP model & file names.

November 30th, 2022

(Login: Ruofan) Added unit tests to cover entire presenter file & test its behaviour.

Student

Shanni: optimized the layout and buttons and add more conditions when student want to click button to make sure the app will not terminated accidentally, with adding lots of toast to reminder the students

Xinyu: Implemented the helper methods for the second part of generating timeline method: generateNumList, generateRep, generateSemList, helperYear, checkPre.

Rebecca: Worked on letting admins edit courses by creating the UI base for it, ported over AdminCreate.java code into it and worked on modifying it to make it work for AdminEdit

Juno: attempted to refactor code for efficiency, decided to keep present version.

December 1st, 2022

Juno - Merged admin branch into login1 (temporary copy of login to protect against merge-based code loss), dealt with 4 merge conflicts.

Also refactored all of the view/delete admin courses code so that the bulk of logic is handled by fragments instead of activities, and created the navigation code connecting rebecca’s add/edit course code to my view/delete course code.

Rebecca - Continued working on AdminEdit. It now reads a course of the user’s choice to edit and fills in the information fetched from the static arraylist into the fragment UI.

Xinyu - finished the rest of the generating timeline – putting all the required courses into different semesters, created a new class called Timeline to store the returning data of generatingTimeline method, and tested my code.

Shanni: did a lot of tests for viewing, adding and deleting past courses and future courses to make sure the app will run as I expect and solve some bugs that may be caused due to invalid inputs.

(Login: Ruofan): tested unit test more & refactored code again due to ambiguous/incoherent instructions from different TAs and the instructor; merged with admin & created navigations between login & admin

December 2nd, 2022

Shanni - Got the latest format in the firebase and started to import the course list from firebase to student part. Created a test branch as a temporary branch in order to test the correctness when implementing the firebase import. Finished implementing importing firebase and translating the data on firebase to the format we need in the student part, instead of modifying too much code we wrote in other student parts (not violating OCP). In order to save data and provide the translated data to other activities and other methods, and due to the feature of reading data asynchronously of realtime firebase, I spent a lot of time on finding a method to ensure the data I obtained from realtime firebase is latest and completest and can be used in other place, and finally solve it successfully! My next step is waiting for the admin part to finish exporting the firebase part and then I will delete my test branch on firebase and then use the real one.

Juno - Started working on fetching data from firebase to be displayed in AdminView (show the admin the list of current courses added in the database)

admin

Rebecca - Debugged both AdminCreate and AdminEdit to ensure that no issues occurred. Implemented logging to Firebase and replaced all local storage functions with server storage functions.

Juno - Troubleshooted retrieving firebase data with Ruofan, successfully implemented displaying data from database, keeping data updated during the Admin session. Started working on deleting courses from the database feature.

(Login: Ruofan): added admin view related files to merged admin & login, started merging with student side as well & added that navigation; helped Rebecca with implementing firebase into the admin side

December 3rd, 2022

Juno - implemented double click button feature, integrated firebase for showing and deleting courses. major github pushing problems, so we just handed off code via zip files (no github commit)

**assumptions made: only 1 admin online and editing at a time**

Student & admin

**There will not be a course that has a nonexistent prerequisite**

Ruofan: helped Juno with implementing firebase into the admin side & debugging for issues

Rebecca: Included Juno’s course-viewing function into the AdminCreate.java and AdminEdit.java files, used it to display a list of prerequisites in their respective UIs.

**In addition to the previous assumption, Admins will not make a course a prerequisite/co-prerequisite of itself. That is, if course A is a prerequisite of course B, then course B will not also be a prerequisite of course A.** The app will try to prevent users from performing such actions, but the updates happen upon fragment refresh and will thus not be able to prevent irresponsible course editing within one instance of a fragment. For example, in the AdminEdit java class, a prerequisite A that was just added to a course B will not be included in the list of exclusions for the *course* A if trying to add prerequisite B. However, this will be accounted for if the fragment is destroyed then recreated.

Xinyu: finalized the format of displaying timeline on app with Shanni, modified generatingTimeline, and fixed some bugs on Android Studio for the student part.

Shanni: Designed the UI and framework of generating timeline and added a lot of restrictions and conditions of input of generating timeline to ensure the function of generating timeline will run as we expected. Tested the generating timeline functions and made sure it works with valid inputs. Cleaned the code and generated the final code of the student part.

December 4 & 5th, 2022

Juno - merged admin\_from\_login2 with login2 to combine firebase integration for admin side and student side, fixed merge issues in three files (issues with git commit, so just handed off final file as a .zip, working on fixing to commit to main today/tomorrow)

Rebecca - Implemented UI fixes to the app, including unclear text and overlapping buttons issues, visual acknowledgement of selecting prerequisites in admin course creation and editing, and renaming fragment titles.

Ruofan - verified that the fina copy does run (tested it), and confirmed criteria of submission (e.g., uploading of SCRUM files, JIRA, code is in main branch)