uOttawa

Faculté de génie
Faculty of Engineering

# Project Report

Student Booking Application

Marcus Dillon - 300113181

Alexandre Poirier – 300002452

Axel Tang – 300164095

Nicholas Mah – 0300255901

Gael Sanfo –8648163

SEG 2105 [Z]

Dr. Omar Badreddin.

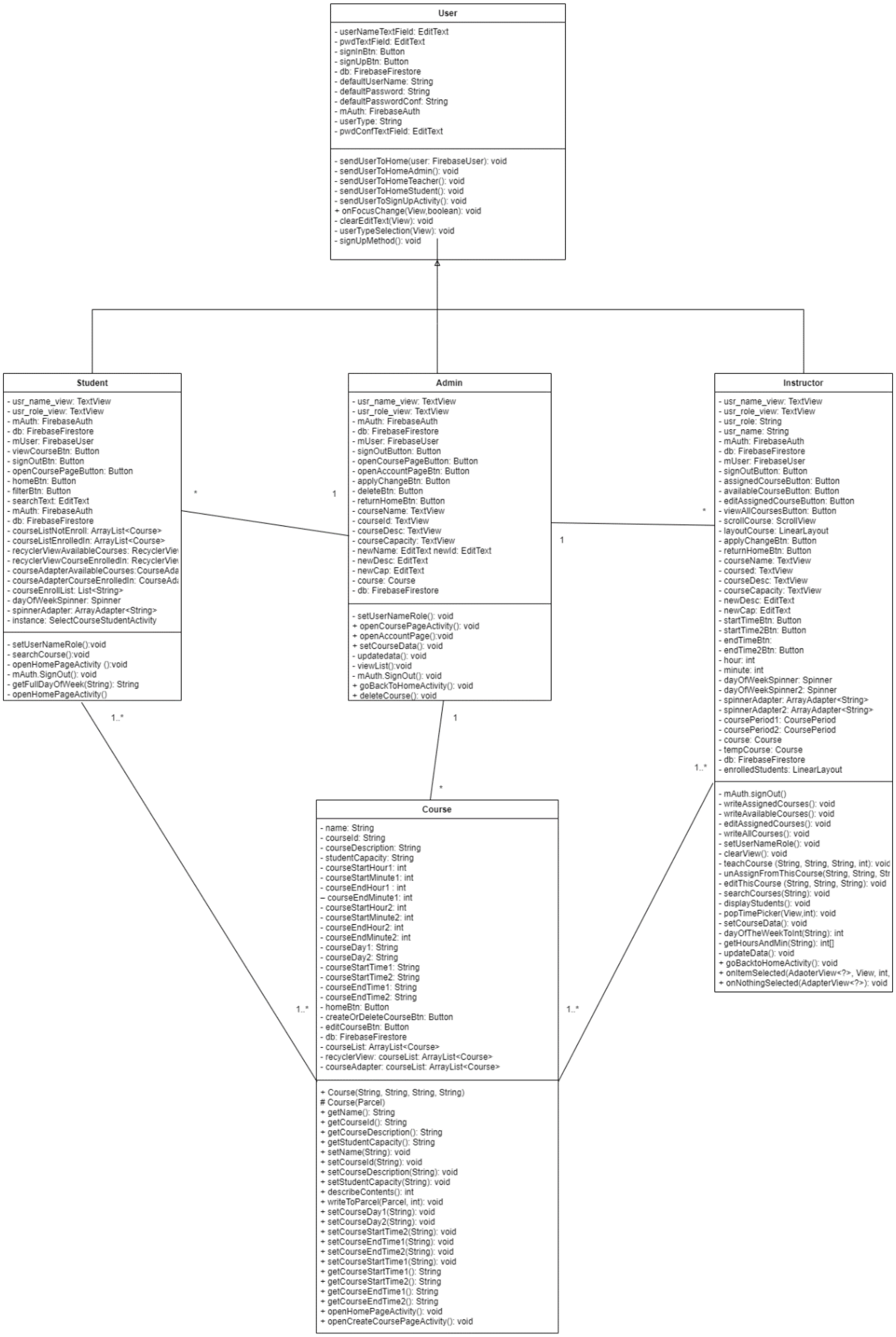
July 31st, 2022

# Contents

# Introduction

This report will provide a summary of the term project for SEG2105, Introduction to Software Engineering. The purpose of this team project was to gain hands-on experience collaborating with a team to build a piece of software. This would allow us to apply our theoretical learning from class in a real setting whilst also learning about the challenges that arise in this type of project. We built the application from the ground up starting with learning how to use Android Studio then proceeding to build our UML diagrams and then slowly implementing blocks of code. The project was divided into three deliverables, with increasing levels of complexity. Each deliverable had several features which had to be implemented by a set deadline. We got the opportunity to practice UML diagrams, Git, JUnit testing, Databases with Firestore, and much more. The application itself constitutes a student course booking app that allowed users to complete various functions related to picking courses. Users were divided into three groups: Students, Instructors, and Admins. Each category has functionality pertaining to only that group of people, for example, only instructors can assign themselves to a course. The application functionality was simple to allow us to focus on other aspects of software engineering such as team management, planning, and collaboration using git. In addition, we were encouraged to implement any additional features that we thought could be of use which allowed us to explore new concepts that reached beyond the scope of the course.

# UML Class Diagram


User
- userNameTextField: EditText
- pwdTextField: EditText
- signInBtn: Button
- signUpBtn: Button
- db: FirebaseFirestore
- defaultUserName: String
- defaultPassword: String
- defaultPasswordConf: String
- mAuth: FirebaseAuth
- userType: String
- pwdConfTextField: EditText
- sendUserToHome(user: FirebaseUser): void
- sendUserToHomeAdmin(): void
- sendUserToHomeTeacher(): void
- sendUserToHomeStudent(): void
- sendUserToSignUpActivity(): void
+ onFocusChange(View,boolean): void
- clearEditText(View): void
- userTypeSelection(View): void
- signUpMethod(): void
Student
- usr_name_view: TextView
- usr_role_view: TextView
- mAuth: FirebaseAuth
- db: FirebaseFirestore
- mUser: FirebaseUser
- viewCourseBtn: Button
- signOutBtn: Button
- openCoursePageButton: Button
- homeBtn: Button
- filterBtn: Button
- searchText: EditText
- mAuth: FirebaseAuth
- db: FirebaseFirestore
- courseListNotEnroll: ArrayList<Course>
- courseListEnrolledIn: ArrayList<Course>
- recyclerViewAvailableCourses: RecyclerVie
- recyclerViewCourseEnrolledIn: RecyclerVie
- courseAdapterAvailableCourses:CourseAda
- courseAdapterCourseEnrolledIn: CourseAd
- courseEnrollList: List<String>
- dayOfWeekSpinner: Spinner
- spinnerAdapter: ArrayAdapter<String>
- instance: SelectCourseStudentActivity
- setUserNameRole():void
- searchCourse():void
- openHomePageActivity ():void
- mAuth.SignOut(): void
- getFullDayOfWeek(String): String
- openHomePageActivity()
*
1
Admin
- usr_name_view: TextView
- usr_role_view: TextView
- mAuth: FirebaseAuth
- db: FirebaseFirestore
- mUser: FirebaseUser
- signOutButton: Button
- openCoursePageButton: Button
- openAccountPageBtn: Button
- applyChangeBtn: Button
- deleteBtn: Button
- returnHomeBtn: Button
- courseName: TextView
- courseId: TextView
- courseDesc: TextView
- courseCapacity: TextView
- newName: EditText newId: EditText
- newDesc: EditText
- newCap: EditText
- course: Course
- db: FirebaseFirestore
- setUserNameRole(): void
+ openCoursePageActivity(): void
+ openAccountPage():void
+ setCourseData(): void
- updatedata(): void
- viewList():void
- mAuth.SignOut(): void
+ goBackToHomeActivity(): void
+ deleteCourse(): void
1
*
Instructor
- usr_name_view: TextView
- usr_role_view: TextView
- usr_role: String
- usr_name: String
- mAuth: FirebaseAuth
- db: FirebaseFirestore
- mUser: FirebaseUser
- signOutButton: Button
- assignedCourseButton: Button
- availableCourseButton: Button
- editAssignedCourseButton: Button
- viewAllCoursesButton: Button
- scrollCourse: ScrollView
- layoutCourse: LinearLayout
- applyChangeBtn: Button
- returnHomeBtn: Button
- courseName: TextView
- coursed: TextView
- courseDesc: TextView
- courseCapacity: TextView
- newDesc: EditText
- newCap: EditText
- startTimeBtn: Button
- startTime2Btn: Button
- endTimeBtn:
- endTime2Btn: Button
- hour: int
- minute: int
- dayOfWeekSpinner: Spinner
- dayOfWeekSpinner2: Spinner
- spinnerAdapter: ArrayAdapter<String>
- spinnerAdapter2: ArrayAdapter<String>
- coursePeriod1: CoursePeriod
- coursePeriod2: CoursePeriod
- course: Course
- tempCourse: Course
- db: FirebaseFirestore
- enrolledStudents: LinearLayout
- mAuth.signOut()
- writeAssignedCourses(): void
- writeAvailableCourses(): void
- editAssignedCourses(): void
- writeAllCourses(): void
- setUserNameRole(): void
- clearView(): void
- teachCourse (String, String, String, int): void
- unAssignFromThisCourse(String, String, Str
- editThisCourse (String, String, String): void
- searchCourses(String): void
- displayStudents(): void
- popTimePicker(View,int): void
- setCourseData(): void
- dayOfTheWeekToInt(String): int
- getHoursAndMin(String): int[]
- updateData(): void
+ goBacktoHomeActivity(): void
+ onItemSelected(AdaoterView<?>, View, int,
+ onNothingSelected(AdapterView<?>): void
1..*
1
*
1..*
1..*
1..*
Course
- name: String
- courseId: String
- courseDescription: String
- studentCapacity: String
- courseStartHour1: int
- courseStartMinute1: int
- courseEndHour1 : int
– courseEndMinute1: int
- courseStartHour2: int
- courseStartMinute2: int
- courseEndHour2: int
- courseEndMinute2: int
- courseDay1: String
- courseDay2: String
- courseStartTime1: String
- courseStartTime2: String
- courseEndTime1: String
- courseEndTime2: String
- homeBtn: Button
- createOrDeleteCourseBtn: Button
- editCourseBtn: Button
- db: FirebaseFirestore
- courseList: ArrayList<Course>
- recyclerView: courseList: ArrayList<Course>
- courseAdapter: courseList: ArrayList<Course>
+ Course(String, String, String, String)
# Course(Parcel)
+ getName(): String
+ getCourseId(): String
+ getCourseDescription(): String
+ getStudentCapacity(): String
+ setName(String): void
+ setCourseId(String): void
+ setCourseDescription(String): void
+ setStudentCapacity(String): void
+ describeContents(): int
+ writeToParcel(Parcel, int): void
+ setCourseDay1(String): void
+ setCourseDay2(String): void
+ setCourseStartTime2(String): void
+ setCourseEndTime1(String): void
+ setCourseEndTime2(String): void
+ setCourseStartTime1(String): void
+ getCourseStartTime1(): String
+ getCourseStartTime2(): String
+ getCourseEndTime1(): String
+ getCourseEndTime2(): String
+ openHomePageActivity(): void
+ openCreateCoursePageActivity(): void

## Team Contribution Table

| Team Member Contribution – Deliverable 1 | | | | |
|---|---|---|---|---|
| Marcus | Alex | Axel | Gael | Nicolas |
| ➢ Admin can create a course (Courses have course code and name)<br>➢ Admin can edit the course code and name of an existing course<br>➢ Admin can delete a course<br>➢ Courses Firestore<br>➢ Video edit | ➢ Can login as admin<br>➢ Can create and login as instructor account<br>➢ Can create and login as student account<br>➢ Admin can delete student and instructor accounts | ➢ UML Class Diagram<br>➢ Welcome screen after authentication | | ➢ Login authentication<br>➢ Firebase for login credentials |

| Team Member Contribution – Deliverable 2 | | | | |
|---|---|---|---|---|
| Marcus | Alex | Axel | Gael | Nicolas |
| ➢ Beatification of the UI<br>➢ Video edit submission | ➢ 5 unit tests related to D1 or D2<br>➢ Once assigned as instructor, they can add course days, hours ,description<br>➢ Instructor can edit course days, course hours, course description and student capacity | ➢ Updated UML class diagram for your domain model.<br>➢ Field Validation | ➢ Instructor can edit course days, course hours | ➢ Instructor can view all created courses<br>➢ Instructor can search for a course by Name and Code<br>➢ Instructor can assign themselves to a course<br>➢ Instructor can un-assign from a course<br>➢ Instructor cannot assign to a course which already has |

| | | | | |
|---|---|---|---|---|
| | | | | an instructor<br>➢ BONUS – Instructor sees a different instructor’s name |

| Team Member Contribution – Deliverable 3 | | | | |
|---|---|---|---|---|
| Marcus | Alex | Axel | Gael | Nicolas |
| ➢ Final report (Title page, introduction, UML, Table contribution, screenshots, lessons learned)<br>➢ Video edit submission | ➢ Student can search for a course by Code, Name and Day of week<br>➢ Student can enroll in a course<br>➢ Student can unenroll<br>➢ Student can see a list of courses that they are enrolled for | ➢ Updated UML class Diagram | | ➢ Students receive an error when trying to enroll in a course with conflicts<br>➢ BONUS – Instructor can see a list of all enrolled students |

## Lessons Learned / Challenges Overcome

This experience proved to be a very good learning lesson on what to and not to do in a software engineering project. Below are some of the key lessons and takeaways which we learned in our journey.

Firstly, everyone has varying skill sets and preferences when it comes to the type of work they are doing. It is important to allow everyone to have a chance to speak their mind when it comes to tasks they are assigned as there is nothing worse than someone who is stuck doing a portion of work they despise. This often leads to poor results and bad morale for the team which could have been easily avoided.

This leads right into the second lesson learned which was not being afraid to ask for help and offer assistance to those in need. It can get quite frustrating when one person is stuck on a portion of work or code for a long time and sometimes a second pair of eyes can really help resolve the problem which is beneficial for the entire team. Everyone in the group has to understand that it is true we all have our separate roles and responsibilities but regardless, we are still a team who is working towards a common goal.

Thirdly, we quickly learned the importance of communication as it is extremely difficult to remain organized with 5 people working on the same project. We had to ensure we had an open line of communication along with a place to store important documents and links. This was accomplished with the creation of a discord group. We had a general channel used for updates and separate channels for varying functions such as important links or deliverable (1-2-3) responsibilities. It is important to note that it only worked so well because we utilized the tools at our disposal. If a team member failed to respond or provide updates on what work they were completed, it became extremely difficult to make progress and move forward in the project.

A fourth lesson was regarding the significance of proper planning. By dividing the work as early as possible, we give each member the most time to complete their portion of the deliverable. This is especially important when one’s work depends on another person as everyone has different schedules as cannot necessarily adapt to all circumstances.

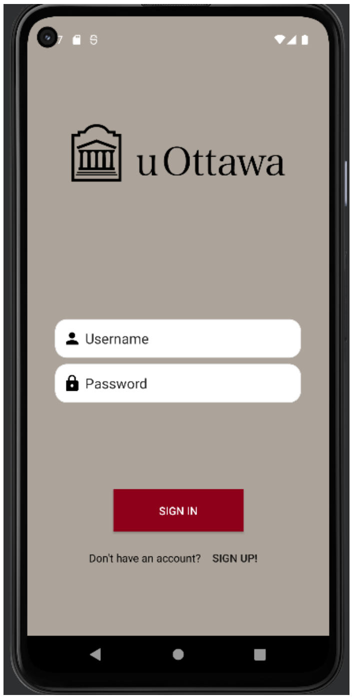
One of the challenges we encountered was that a member of the group didn’t have time to complete their portion of the deliverable as they had conflicting priorities. After making this known to the entire team, we worked something out to ensure that the work would get done regardless whilst keeping an equal division of labor. This was challenging as we often had conflicting priorities which made it difficult, however, we still made it work.

Another challenge we had to overcome was that no one had previous experience using Android Studio. This meant we had to dedicate more time to learning the software and fixing potential issues. This emphasized the importance of the lessons learned above as these challenges are only amplified when everyone is a beginner at using the software required to develop our application.

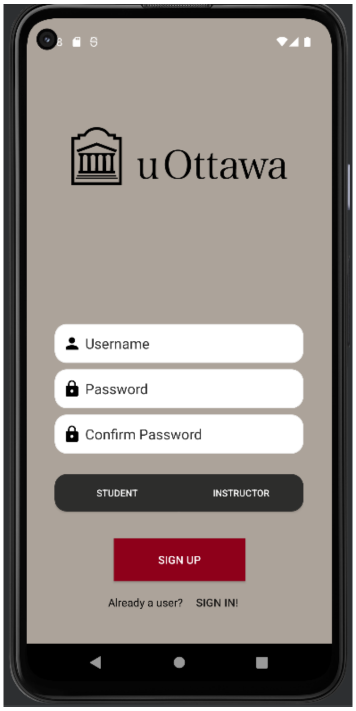
In conclusion, this project was a perfect way for us students to learn the challenges that come along with software development. Gaining theoretical knowledge is one thing but putting these concepts into practice in a real-world scenario helped reinforce our learning and ultimately made us much better developers and future software engineers.

# Screenshots

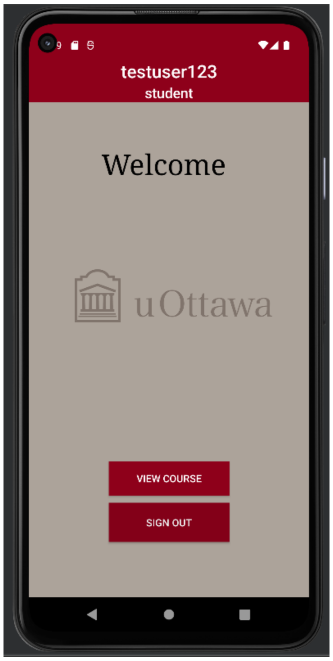
## Login Screen



## Sign up screen



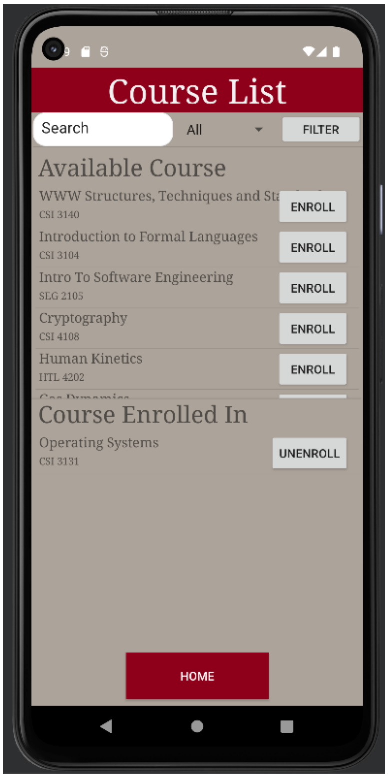
## Welcome – Student



testuser123
student

Welcome

## Course List - Student



Course List

Search All FILTER

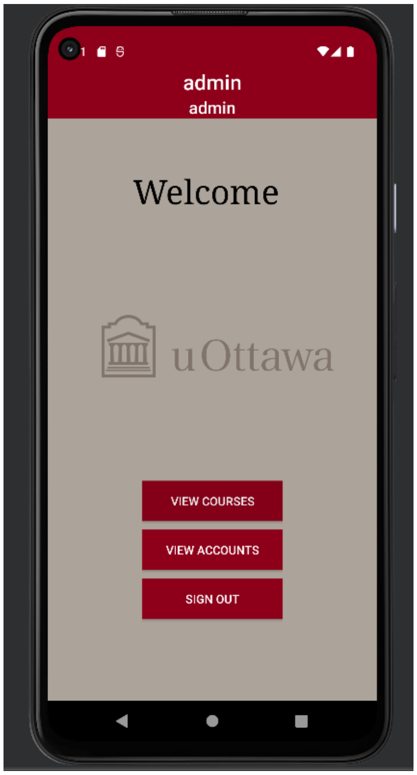
Available Course

| | | |
|---|---|---|
| WWW Structures, Techniques and St | CSI 3140 | ENROLL |
| Introduction to Formal Languages | CSI 3104 | ENROLL |
| Intro To Software Engineering | SLG 2105 | ENROLL |
| Cryptography | CSI 4108 | ENROLL |
| Human Kinetics | HTL 4202 | ENROLL |

Course Enrolled In

| | | |
|---|---|---|
| Operating Systems | CSI 3131 | UNENROLL |

## Welcome – Admin

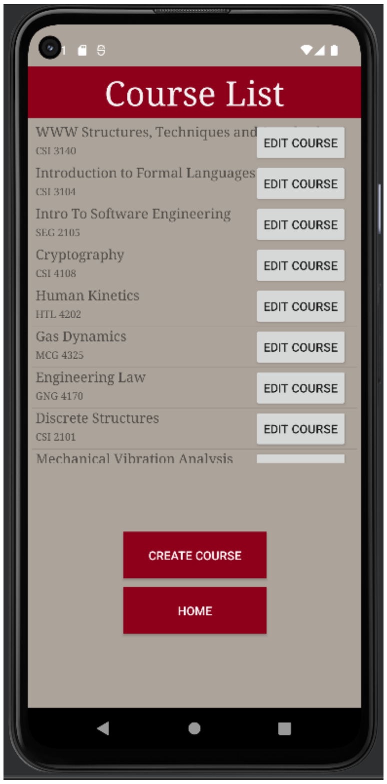


admin

admin

Welcome

## Course List – Admin



Course List

WWW Structures, Techniques and
CSI 3140

Introduction to Formal Languages
CSI 3104

Intro To Software Engineering
SEG 2105

Cryptography
CSI 4108

Human Kinetics
HTL 4202

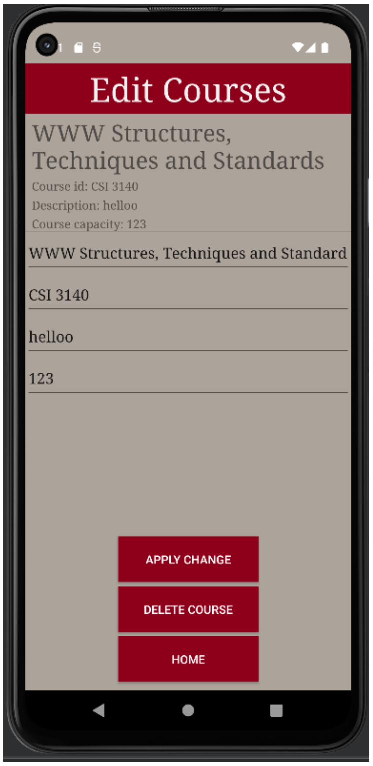
Gas Dynamics
MCG 4325

Engineering Law
GNG 4170

Discrete Structures
CSI 2101

Mechanical Vibration Analysis

## Edit Course – Admin



Edit Courses

WWW Structures, Techniques and Standards

Course id: CSI 3140

Description: helloo

Course capacity: 123

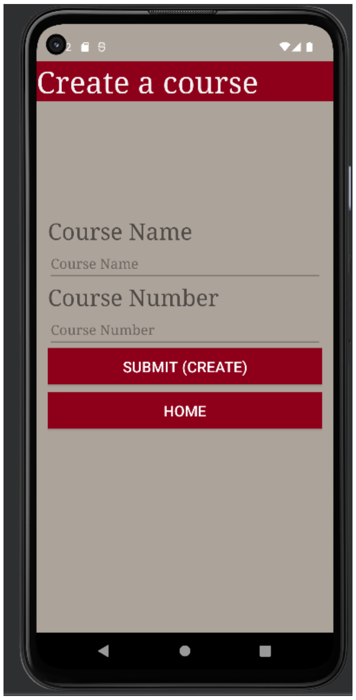
WWW Structures, Techniques and Standard

CSI 3140

helloo

123

## Create Course – Admin



Create a course

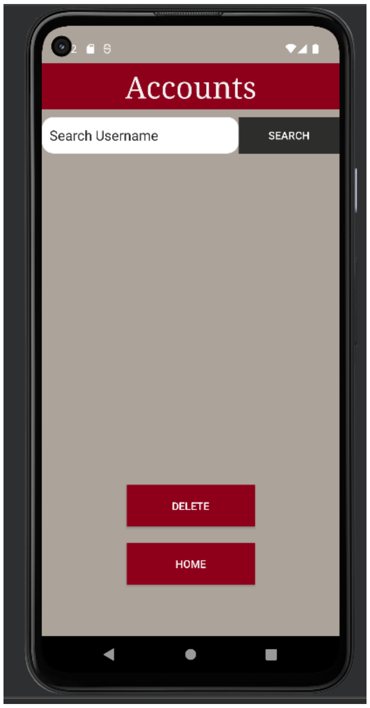
Course Name

Course Name

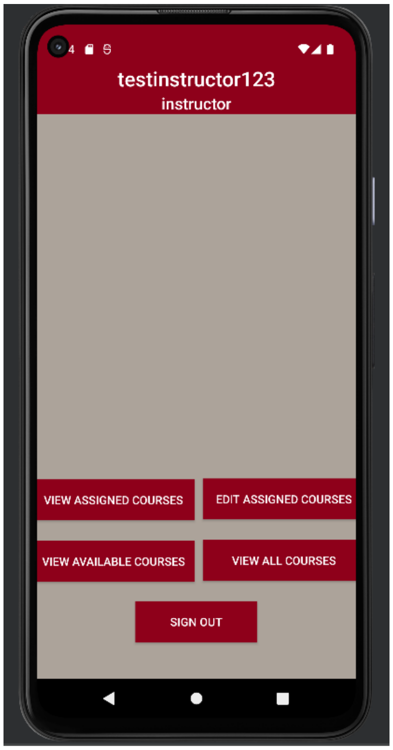
Course Number

Course Number

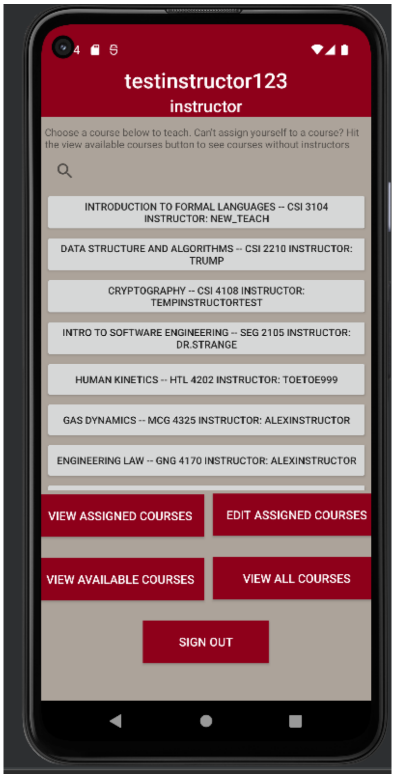
## View Account – Admin



## Welcome – Instructor



## View Courses – Instructor



testinstructor123

instructor

Choose a course below to teach. Can't assign yourself to a course? Hit the view available courses button to see courses without instructors

INTRODUCTION TO FORMAL LANGUAGES -- CSI 3104 INSTRUCTOR: NEW_TEACH

DATA STRUCTURE AND ALGORITHMS -- CSI 2210 INSTRUCTOR: TRUMP

CRYPTOGRAPHY -- CSI 4108 INSTRUCTOR: TEMPINSTRUCTORTEST

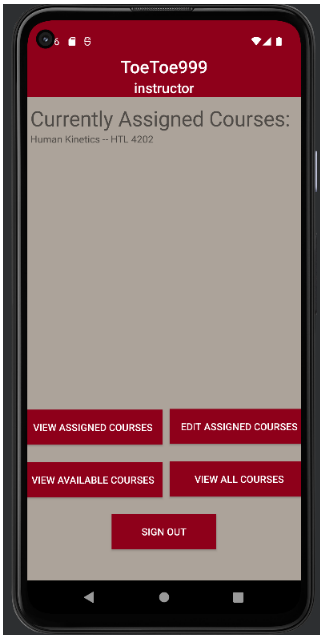
INTRO TO SOFTWARE ENGINEERING -- SEG 2105 INSTRUCTOR: DR.STRANGE

HUMAN KINETICS -- HTL 4202 INSTRUCTOR: TOETOE999

GAS DYNAMICS -- MCG 4325 INSTRUCTOR: ALEXINSTRUCTOR

ENGINEERING LAW -- GNG 4170 INSTRUCTOR: ALEXINSTRUCTOR

## View Assigned Courses – Instructor



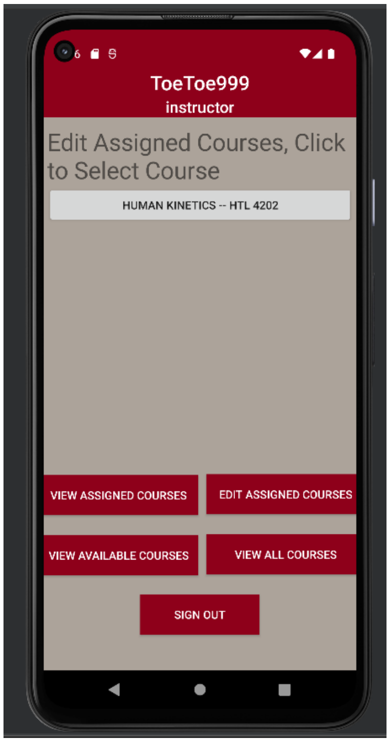
ToeToe999

instructor

Currently Assigned Courses:

Human Kinetics -- HTL 4202

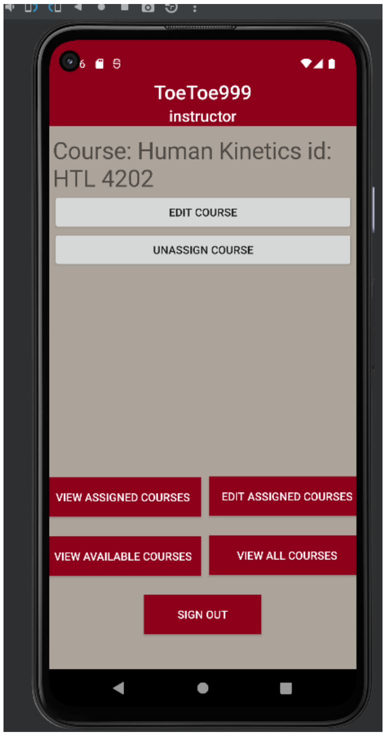
## Edit Assigned Courses – Instructor



ToeToe999

instructor

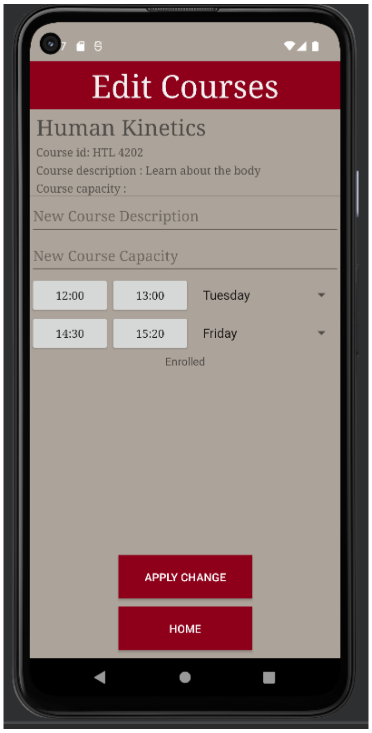
Edit Assigned Courses, Click to Select Course



ToeToe999

instructor

Course: Human Kinetics id: HTL 4202



Edit Courses

Human Kinetics

Course id: HTL 4202

Course description : Learn about the body

Course capacity :

New Course Description

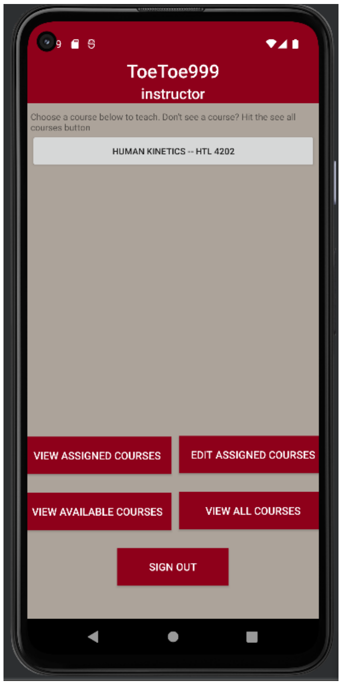
New Course Capacity

12:00 13:00 Tuesday

14:30 15:20 Friday

Enrolled

## View Available Courses – Instructor



ToeToe999

instructor

Choose a course below to teach. Don't see a course? Hit the see all courses button