GrangeMobile App Report

Student Name	Karan Gupta
Student ID	D22124440
@ Email	D22124440@mytudublin.ie
Module	Social Media Applications | MED9027: 2022-2023
Last edited time	@March 13, 2023 5:39 PM
Created time	@March 12, 2023 4:25 PM

1. User Definition

1.1 Selected User - *The Lecturer*

For the GrangeMobile Application, I have selected the lecturer to be the primary user of this application because lecturers are the employees of the university and they ought to use this app more often than any other user.

1.2 Elements of User Definition

User Persona

John Doe is a recently employed professor at TU Dublin. It is his first time teaching in a large university with a complex organizational structure. As one week has passed by he is finding it difficult to track his modules and courses. Management has been training him to better organize his schedule but it is getting very difficult to manage everything on an Excel spreadsheet.

Moreover the classes he has been assigned are complaining that they are not updated of any kind of changes well in advance. All these organisational issues are hampering his ability to teach the subject he was hired for. He is seriously considering a job switch to a smaller college where he can focus more on the subject than the schedule management. Unless he finds a software application that helps him view all his courses, organize his schedule and contact his classes without any fuss.

User Goals

- Become more informed about his courses/modules
- Better organize his time at the university
- Keep his classes informed of any updates or announcements

User Frustrations

- Not enough time/energy to create schedule using traditional tools
- Loses track of class-specific announcements via individual emails
- Was hired to be subject expert and not for managing their classes

> 💡 The GrangeMobile App aims to solve these problems and offer the lecturers a simpler way to view and manage their schedules, and contact their class students without using any traditional tools.

User Scenario

John Doe, our previously defined user, receives an email from college administration with instructions for his first day in college. Here he is asked to download the GrangeMobile App which will help him in organising and contacting his classes. He starts exploring the application and discovers that the modules and courses that were assigned to him are already available to him on the app. He is relieved that he won't have to put in too much effort into creating his time table.

Once he is done exploring his schedule, he discovers that he can also contact the students of his classroom within the app itself. He is finally relieved that much of the management work is taken off his hands and he can now focus on making his presentation for tomorrow's first class. He finished off the presentation, takes a look at his schedule for one last time and sleeps early without any unnecessary stress.

---

3. Use Case No. 1

3.1 Use Case: Lecturer Wants to View the Schedule.

In the first use case I have considered a situation where the lecturer, John Doe, simply wishes to view his time table for the upcoming classes such that he can prepare his content accordingly.

System

The system involved in this use case will be the online scheduling system prepared by the university which helps students, lecturers and administrators to view and update the time tables according to availability of lecturers, duration of classes, location of classes, and corresponding clashes with each other. The scheduling system is assumed to be prepared in advance by the administrators before the term begins so that concerned parties won't have any conflicts. The GrangeMobile App will be a part of this system which will access information from the system and display it to the lecturers.

Primary Actor

The primary actor who initiates this particular use case will be the lecturer, John Doe.

Secondary Actor

The secondary actors who are involved in this use case and interact with the app will be the databases of schedules, modules and courses which were created and updated by the IT administrators and management professionals in the university.

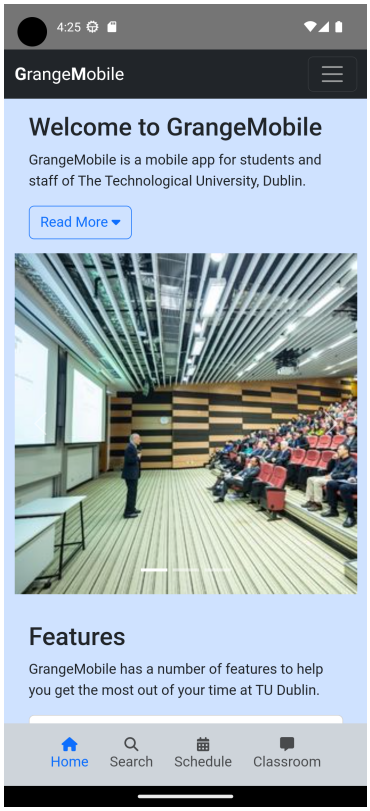
Pre-conditions

The lecturer has already logged in using his secure credentials into the app and has sufficient understanding of courses and schedules from his past experience. He has enough digital literacy to use a simple mobile application. Lecturer has no need nor authority to alter his schedule without prior permissions from the management.

Post-conditions

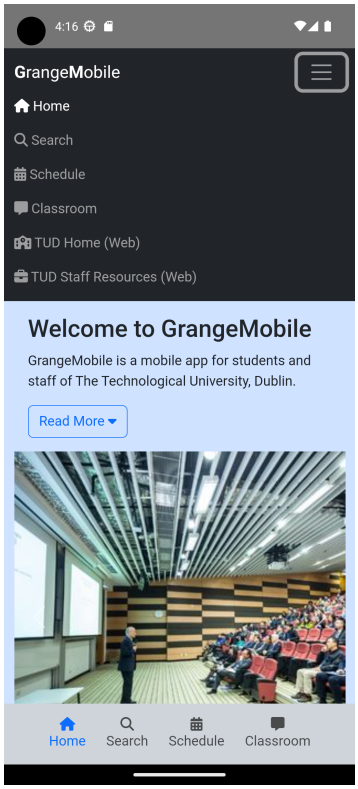
The lecturer has successfully viewed his schedule in the app and possibly transferred the information to his personal native Google or Apple calendar such that he receives timely reminders or notifications.

3.2 Behaviour and Data



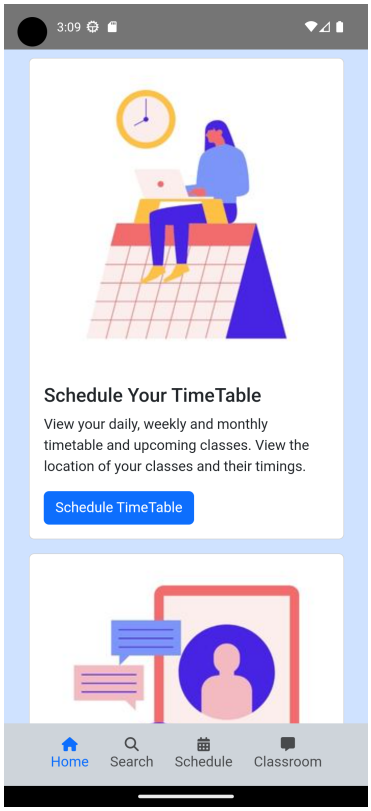
Home Page

1. John opens the app and lands on the ‘Home’ page. Here he explores the features of the page.



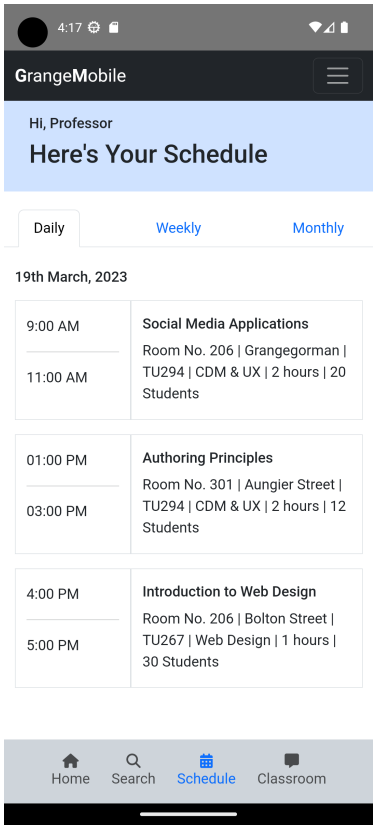
Home Page Top Nav Bar

2. John opens the top navigation bar and clicks on schedule to enter his schedule.



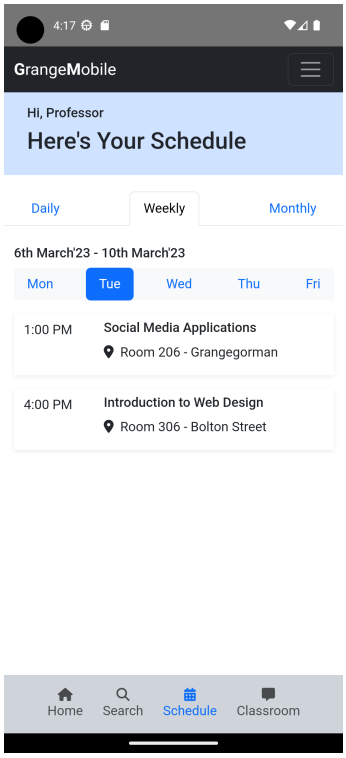
Home Page Feature Cards

3. John can also scroll down on the home page to visit the link to the schedule page.



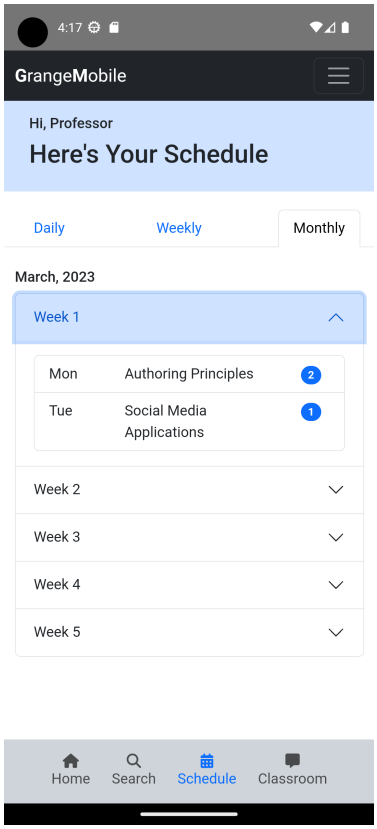
Schedule Page - Daily

4. John lands on schedule page which shows the current day's schedule with time slots.



Schedule Page - Weekly

5. John switches to weekly view using the tabs and views the schedule for Tuesday in the current week.



Schedule Page - Monthly

6. John switches to monthly view using the tab and opens the week 1 accordion bar to view the week's schedule.

The data for the schedule page is fetched from an external JSON file with nested JSON objects and arrays divided according to the daily, weekly and monthly key/value pairs. The data is fetched from this JSON file using a jQuery AJAX Call and is inserted into the DOM under the corresponding tab using template literals, `$.each()` and `$.map()` methods. Below is the condensed form of the `schedule.json` file available in the app source bundle.

```
{
    "daily":[
        {
            "id": 1,
            "moduleName":"Social Media Applications",
            "moduleId":"MED9027",
            "moduleRoom": "206",
            "moduleLocation":"Grangegorman",
```

```
            "startTime":"9:00 AM",
            "endTime":"11:00 AM",
            "courseID":"TU294",
            "courseName":"CDM & UX",
            "moduleDuration": 2,
            "moduleStudents": 20
        }, {...}, {... }
    ],
    "weekly":[
        { "monday":[{...}, {...}]},
        {"tuesday":[{...}, {...}]},
        {"wednesday": [{...}, {...}]},
        {"thursday":[{...}, {...}]},
        {"friday":[{...}, {...}]}
    ],
    "monthly":[
        {"week1":[{...}, {...}]},
        {"week2":[{...}, {...}]},
        {"week3":[{...}, {...}]},
        {"week4":[{...},{...}]},
        {"week5":[{...},{...}]}
    ]
}
```

3.3 Prioritised Behaviour List

1. User opens the app and lands on the home page.
2. User goes through the home page and navigation panels to search for schedule page.
3. User clicks on the schedule feature card on the home page and lands on schedule page.
4. User is shown the current day’s schedule in a tabular format with all the required information such as class location, duration, course id, room no., and they are distributed according to time durations.
5. User now switches to the weekly tab on the top nav bar because he wants to see what all classes he has in the current week. Here user can switch between different days using the top navigation tabs to view their specific schedules.
6. User now wants to get an overview of his schedule throughout the month. So he switches to the monthly tab on top nav bar. Here he is presented with an accordion type view with separate weeks for the specific month. User taps on week 1 and the accordion opens up with a brief overview of the classes on different days of the week.

7. User is satisfied with the schedule and can now focus on creating valuable class content.

4. Use Case No. 2

4.1 Use Case: Lecturer Sends a Message to their Students.

In the second use case, John Doe, our lecturer, wants to send an individual message to one of the students from his class ‘Social Media Applications’. He also discovers that since the student is a new admission, he is not available on the student list. The lecturer will first add the student to the student list and then send the message to all students.

System

The GrangeMobile App will the primary system here and it will work with the mobile’s native e-mail application like Gmail or Outlook. The system will be created by the administrator for each module such that lecturers can easily contact their students without going to a third-party email system. This system would help lecturers to segregate individual courses and keep a repository of all the messages they’ve sent to the class as a whole or individually.

Primary Actor

The primary actor who initiates this particular use case will be the lecturer, John Doe.

Secondary Actor

The secondary actors who are involved in this use case and interact with the app will be the databases of student data and a storage of sent messages. The student database shall be updated by the administrator and the messages database will be updated as the lecturer sends messages to the students.

Pre-conditions

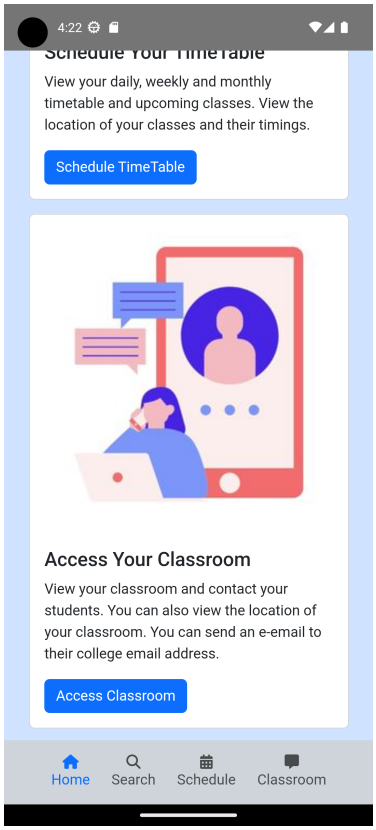
The lecturer is already inside one of his modules, Social Media Applications. There is a preloaded list of students which is displayed in the app. For the sake of demonstration, I’ve created a preloaded database of previously sent messages assuming the lecturer has already sent a couple of messages to the class students

in the past. For this use case, I've assumed that the lecturer can only send messages and cannot receive any messages from students.

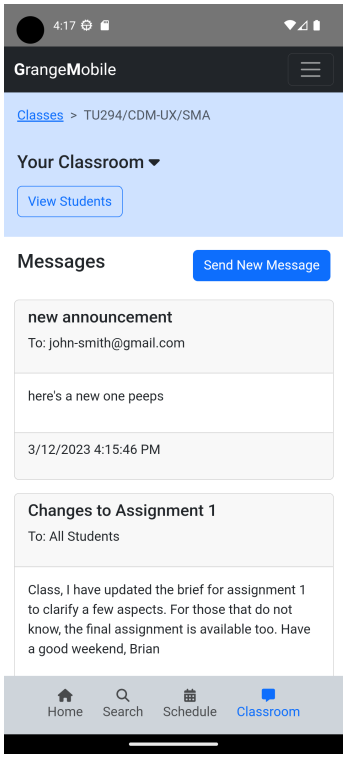
Post-conditions & Assumptions

For the proof-of-concept, once the lecturer hits the send message button, the message doesn't get delivered to the students via the native e-mail application since that was out of the scope of this project. There is a work around where the lecturer can tap on the e-mail link from the 'Your Students' list which will take them to their native email application.

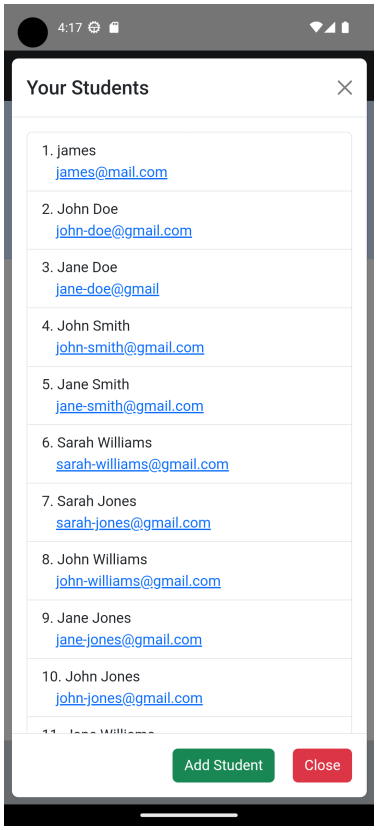
4.2 Behaviour and Data



Home Page



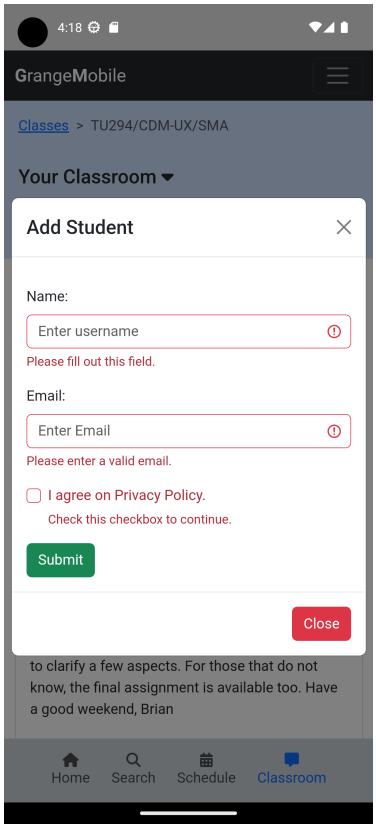
Classroom Page



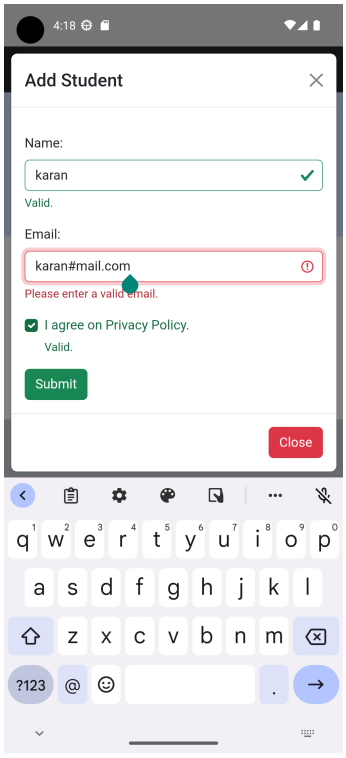
Your Students List Modal Box

1. John opens the app, lands on the home page, scrolls down to the classroom feature card and clicks on 'Access Classroom' button.
2. John lands on the specific Classroom page where he can view the past messages. He taps on 'View Students' button to access the student list.
3. John lands on a 'Your Students' modal box with a list of all students with their names and emails. He taps on 'Add Student' button to

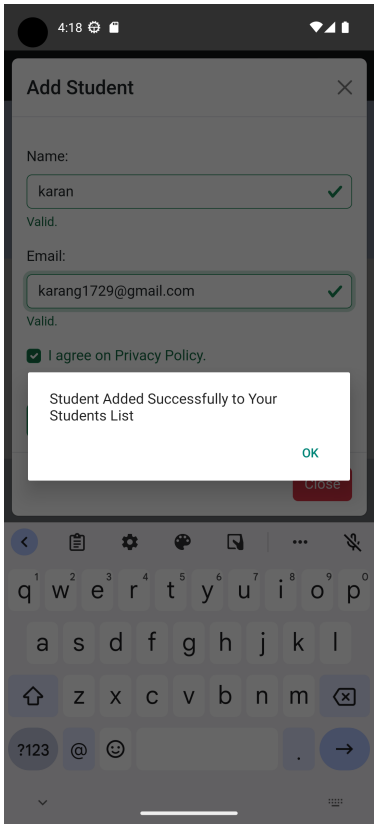
add a new student in list.



Add Student Form Modal Box



Add Student Form Validation

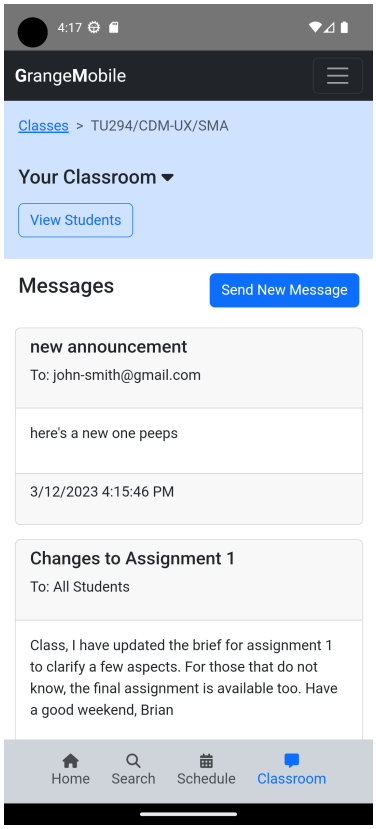


Add Student Success Alert

4. John lands on ‘Add Student’ modal form where needs to enter the student’s name and email address and check privacy policy.

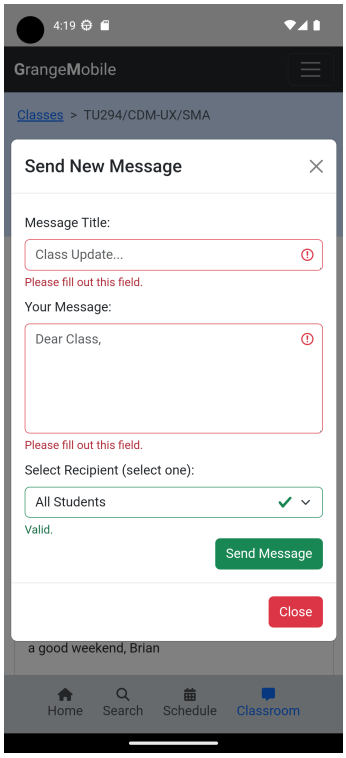
5. John enters the wrong email format and is presented with error messages which is indicative of proper form validation.

6. John enters the correct information in the form and taps on the ‘Submit’ button which displays a success alert message on the screen.



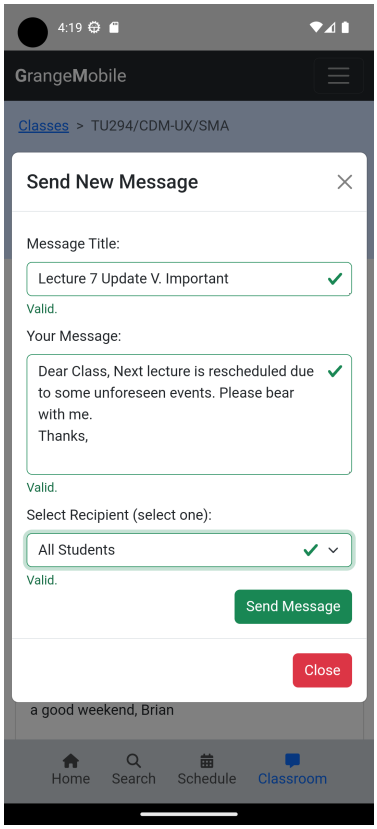
Classroom Page

7. John exits the modal box and lands back on the classroom page. To send a new message to ‘All Students’, he taps on ‘Send New Message’ button.



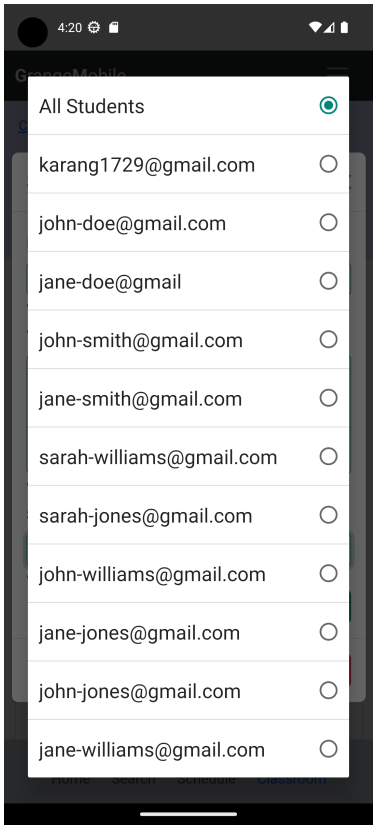
Send New Message Modal

8. John lands on a new modal box form where he needs to enter the message title, message and select whom to send.

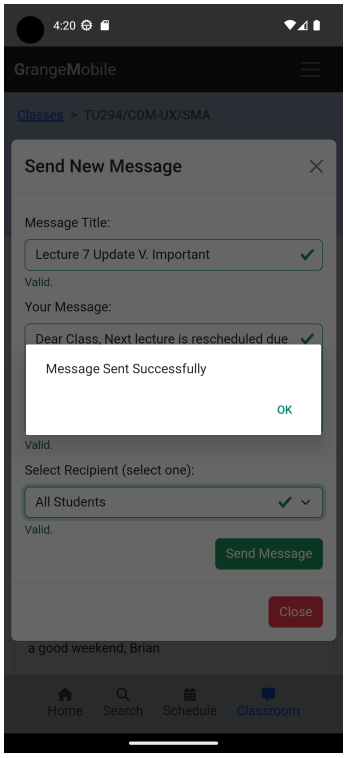


Send New Message Form

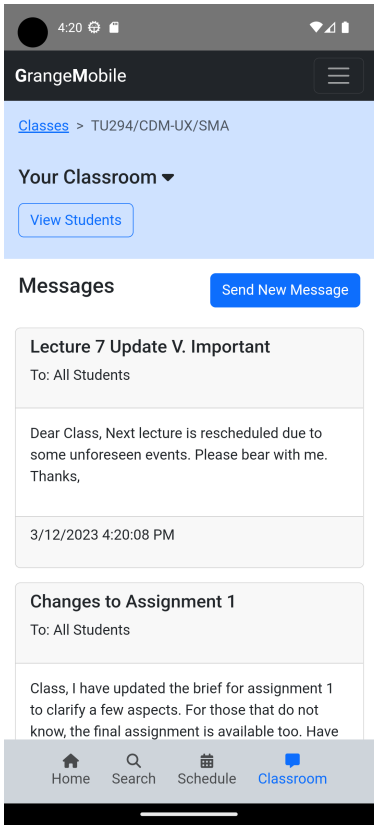
9. John enters the information and gets proper form validation. He taps on recipient dropdown box to select whom to send the message.



Student List in Message Form



Send New Message Success



Updated Classroom Page

10. John is presented with a list of student email IDs with the newly added student from the previous steps. He selects ‘All Students’.
11. John completes the form and taps on ‘Send Message’ giving an alert message for message success. He clicks on ‘OK’ to close the form.
12. John lands on the classroom page where the sent message is displayed at the top of messages list with the time stamp.

Data

For the data, I have created multiple JSON files which are fetched using a jQuery ajax calling using the ‘GET’ method.

For STEP 3, the student list is displayed using the `students.json` file.

```
[
    {
        "id": 1,
        "name": "John Doe",
        "email": "john-doe@gmail.com"
    },
    {
```

```
        "id": 2,
        "name": "Jane Doe",
        "email": "jane-doe@gmail"
    }, {...}
]
```

```
$.getJSON("students.json", function (data) {
    $.each(data, function (i, student) {
      $("#student-list").append(`
      <li class="list-group-item">
        ${student.name} <br />    
         <a href="mailto:${student.email}">${student.email}</a>
       </li>
      `);
    });
  });
```

The students associative array is fetched using `$.getJSON()` method and looped through using `$.each()` method and further added into the parent element of ordered list using `$(selector).append(``)` method which uses template literals and allows us to access the each key/value pair from the individual object in the array.

For STEP 2, the data is fetched from a `messages.json` file which stores all the previously sent messages and displays them as bootstrap cards under the messages section on the classroom page. The jQuery methods are similar to the ones described previously.

```
[
    {
        "id": 1,
        "messageTitle": "Changes to Assignment 1",
        "messageBody": "Class, I have updated the brief for assignment 1 to clarify  a
few aspects. For those that do not know, the final assignment is available too. Have a
good weekend, Brian",
        "messageDate": "24 February 2023",
        "messageTime": "5:23 PM",
        "messageRecipient": "All Students"
    },

    {
        "id": 2,
        "messageTitle": "Lecture 3 Video",
        "messageBody": "Class,the video for lecture 3 on user stat gathering technique
s is up under the lecture 3 content section I will also remind you all that Thursday's
class is a workshop on paper prototyping. Kind regards, Brian",
        "messageDate": "20 February 2023",
        "messageTime": " 12:58 PM",
        "messageRecipient": "All Students"
```

```
    }, {...}
]
```

4.3 Prioritised Behaviour List

1. John opens the app and is greeted by a visually appealing home page with a variety of features available to him. After taking a moment to scan the page, his eyes drift towards the classroom feature card which immediately catches his attention. He scrolls down to take a closer look at the card and reads about the various benefits of using the classroom feature. Intrigued, he clicks on the 'Access Classroom' button, excited to see what the feature has to offer.
2. John navigates to the Classroom page that he needs. Once there, he can easily view all past messages for the class. In order to get a better understanding of the students in the class, he clicks on the ‘View Students’ button which will take him to a page where he can see a list of all the students enrolled in the class. From there, he can further explore each student's profile to gain more insight into their background and academic performance. This feature is incredibly useful for educators who want to stay up-to-date on their student's progress and ensure that every student is receiving the support they need to succeed.
3. He navigates to the 'Your Students' section and is presented with a modal box that displays a list of all his current students, including their names and email addresses. After reviewing the list, John realizes that he needs to add a new student to his class. To do so, he clicks on the 'Add Student' button, which opens up a form where he can input the new student's information, such as their name, email, and other relevant details. Once the information has been entered, John clicks 'Submit' and the new student is added to the list. By keeping his student list up-to-date, John can effectively manage his class and ensure that all his students receive the support they need to succeed.
4. After John exits the modal box, he finds himself back on the classroom page. Here, he can perform a variety of actions, such as sending a new message to all students. To do this, he simply taps on the ‘Send New Message’ button, which will allow him to compose and send a message with ease. This is a great feature that enables John to easily communicate with his students and keep them updated on important information related to the classroom. By using this feature, John can save valuable time and streamline his communication with students, ensuring that everyone is on the same page and working towards a common goal.