**CS 442**

**Mobile Application Development with Android**

**Project**

**Team 4**

**TA-Helper**

***Team Members:***

***Aditya Jadhav***

***Mohammad Shethwala***

***Sowmya Parameshwara***

***Ullas Aithal Kota***

**Table of Contents**

[1. General Description 2](#_Toc468030770)

[2. Feature List 3](#_Toc468030771)

[3. Elevator Pitch 3](#_Toc468030772)

[4. UI Design/Screen drawings and their transition flow 3](#_Toc468030773)

[4.1. Layouts: 3](#_Toc468030774)

[4.1.1. Login Screen: 3](#_Toc468030775)

[4.1.2. Main Screen: 3](#_Toc468030776)

[4.1.3. Courses Screen: 3](#_Toc468030777)

[4.1.4. Add Course Screen: 4](#_Toc468030778)

[4.1.5. Broadcast Email Screen: 4](#_Toc468030779)

[4.1.6. Student List Screen: 4](#_Toc468030780)

[4.1.7. Student Screen: 4](#_Toc468030781)

[4.1.8. Import Student Data: 4](#_Toc468030782)

[4.1.9. Export Student Data: 4](#_Toc468030783)

[4.1.10. Modules Screen: 4](#_Toc468030784)

[4.1.11. Add Modules Screen: 4](#_Toc468030785)

[4.1.12. Manage Assignments Screen: 4](#_Toc468030786)

[4.1.13. Assignment Split Screen: 4](#_Toc468030787)

[5. Use-Case Diagram 6](#_Toc468030788)

[6. Module Dependency Diagram 6](#_Toc468030789)

[7. Work Assignments and Status 7](#_Toc468030790)

[7.1. Phase 2 - Week 1 7](#_Toc468030791)

[7.2. Phase 2 - Week 2 8](#_Toc468030792)

[7.3. Phase 2 - Week 3 11](#_Toc468030793)

[8. Test Cases 15](#_Toc468030794)

[9. Instructions for testing the app: 33](#_Toc468030795)

[10. Important Information 34](#_Toc468030796)

1. General Description

**TA helper** app is mainly aimed for TAs to manage student affairs related to academics, assignments and grades. This app will serve as an all-in-one package for TAs to maintain up to date records related to academics and subjects. This app also allows communicating announcements, grades etc to the students via email.

1. Feature List

1. Maintain in-class assignment scores.

2. Maintain homework/assignment grades.

3. Maintain exam scores.

4. Maintain project scores.

5. Take the formula and auto calculate final grade for all.

6. Send broadcast email of notifications to all the students through the app.

7. TA can use app to handle multiple classes.

8. See in-class assignments scores or quiz scores for each student.

9. Export all this data to excel or where ever needed for it to be easily available in blackboard.

10. Import student list to the app’s database.

11. Generate random groups of students for the project.

12. View the complete list of students for a subject.

13. Add splits to any module (assignments, project, in class assignments) as per requirement.

14. Send notification to TA after adding the TA to the course.

1. Elevator Pitch

We gave the elevator pitch to Radha (TA) and explained her all the functionalities we intended to include in our app. We also took some feedback related to the topic and added some new requirements to the previous requirements. The TA was convinced with our pitch and approved of the idea.

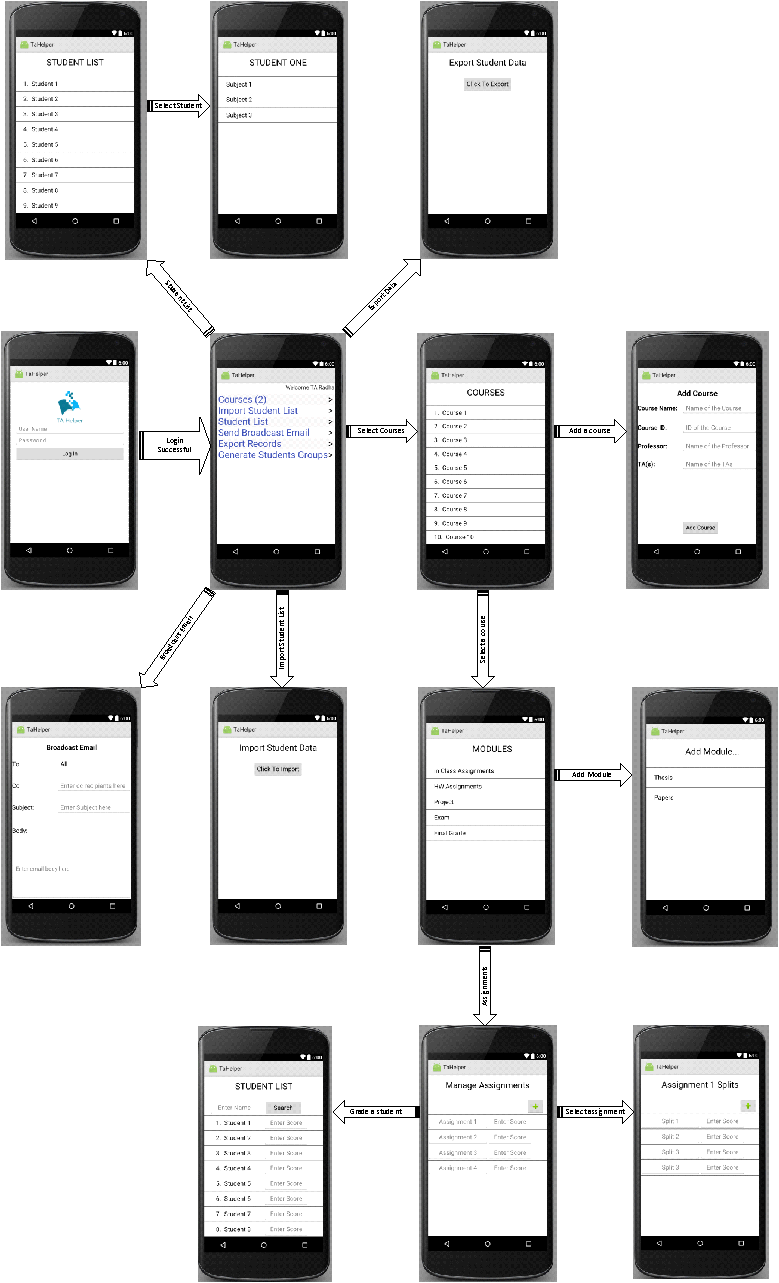
1. UI Design/Screen drawings and their transition flow

The screen drawings and their transition flow is shown on the next page. All the screens we want to use in our app are covered in the Figure 4.1

* 1. Layouts:

Details of all layout screens shown in Figure 4.1

* + 1. Login Screen:   
       Presents login option for TAs
    2. Main Screen:   
       Contains all the modules related to academics. Ex: Courses, Student List, Send Broadcast email, Export Records, Generate Student Groups
    3. Courses Screen:   
       List the courses of the TA
    4. Add Course Screen:   
       Add a new course for the TA to evaluate
    5. Broadcast Email Screen:   
       Send broadcast to email to all the students along with members listed in the cc field
    6. Student List Screen:   
       List the name of students enrolled in the course
    7. Student Screen:   
       List the subjects registered for the sem by that student
    8. Import Student Data:   
       Import the list of students enrolled for the course
    9. Export Student Data:   
       Export various details of the student
    10. Modules Screen:   
        Lists various tasks that can be performed on the TA
    11. Add Modules Screen:   
        Add a module from a pre defined set of modules
    12. Manage Assignments Screen:   
        Manage in class / homework assignments of student(s)
    13. Assignment Split Screen:   
        Define the marks split up for the assignment (Ex step 1-6: 4 marks)



**Fig 4.1**

1. Use-Case Diagram

C:\Users\Mohammed\Google Drive\IIT Semester 2\CS 442 Mobile Apps Dev\Project\TA Helper Use Case Diagram v2.png

**Fig 5.1**

1. Module Dependency Diagram



1. Work Assignments and Status
   1. Phase 2 - Week 1

|  |  |  |
| --- | --- | --- |
| Assign To | Work Desc | Status |
| Ullas, Aditya, Mohammed | Proof of Concept for Firebase, it’s a mobile platform provides nosql database which we can access from android | Completed, Able to read, write data from android |
| Aditya, Sowmya | Setting up the git repository to maintain code synchronization | Completed |
| Aditya | Design layouts for Login Activity, Manage Courses Activity, Manage Assignments Fragment, Manage Assignments Splits Fragment and Generate Students Groups Activity. | Completed |
| Design use case diagram of the Application. | Completed |
| Integrate Firebase into Application as a whole. | Completed |
| Set up waffle io account which tracks the development work and issues inside the github. | Completed |
| Design and created module dependency design diagram for the App. | Completed |
| Implemented the feature of Google Sign In into the App. User is able to sign in into the App. User Details such as name, email id, profile pic are updated into the firebase from the google account. | Completed |
| Implemented the feature of displaying user details on the home activity. | Completed |
| Implemented Log out feature which enables the user to log out from the application | Completed |
| Mohammed | Updating layout files for student list | Completed |
|  | Added fragment, activity and adapter classes for student list | Completed |
| Implemented activity “StudentListActivity” to display list of all the students | Completed |
| Used StudentListFragment and StudentListAdapter to implement the Student List functionality. | Completed |
| Used the class file StudentEntity for storing Student objects | Completed |
| Ullas | Implemented activity “CourseActivity” to handle course related functionalities | Completed |
| Implemented module for Course Addition | Completed |
| Implemented module to display the list of courses | Completed |
| Sync newly added course details to firebase | Completed |
| Retrieve list of courses and course related details from firebase and display it in the app | Completed |
| Connected modules Course Addition and Course List view using fragments | Completed |
| Implemented layout files for List view, custom adapter, Fragment for the modules | Completed |
| Redesigned existing layouts | Completed |
| Create class files for Student and Course entity | Completed |
| Sowmya | Designed layouts for Module List Activity, Module List Fragment, Add Module Activity, Add Module Fragment, Edit/Delete Module Activity, Edit/Delete Module Fragment and Module List Item Adapter. | Completed |
|  | Implemented Module List Activity, Fragment and Adapter for showing the list of Modules. | Completed |
|  | Stored default set of Modules in Firebase and retrieved from there during initial load. | Completed |
|  | Implemented Add Module Activity and Fragment for adding new Module Dynamically. | Completed |
|  | Updated firebase with the newly added module, and triggered update in the module list view. | Completed |
|  | Implemented Edit/Delete Module Activity and Fragment for editing and deleting existing module. | Completed |
|  | Deleted module from firebase and updated module list view. | Completed |
|  | Edited module in firebase and updated module list view. | Completed |
|  | Implemented Activity transition from Module List activity to add module activity. And from add module activity to module list activity. | Completed |
|  | Implemented Activity transition Module List activity to edit/delete module activity. And from edit/delete module activity to module list activity. | Completed |
|  |  |  |

* 1. Phase 2 - Week 2

|  |  |  |
| --- | --- | --- |
| Sowmya | Designed layouts for manage assignment fragment, manage assignment activity, add new assignment with splits fragment, add new assignment with splits activity, edit/delete assignment activity, edit/delete assignment fragment, add assignments item layout, manage assignments item layout. | Completed |
| Implemented manage assignment list activity, fragment and adapter for showing the list of assignments in a module. | Completed |
| Fetched details for the assignments in a module from firebase. | Completed |
| Implemented add new assignment fragment,activity. Implemented splits for assignment score using list item adapter. | Completed |
| Added all newly added assignments into firebase. | Completed |
| Implemented edit/delete assignment fragment,activity. Where edit delete is supported for both assignment details and split details. | Completed |
| Updated firebase with edit/delete actions on assignments and split. | Completed |
| Implemented activity transition from modules to assignments list. | Completed |
| Implemented activity transition from assignment list to add new assignment. | Completed |
| Implemented activity transition from assignment list to edit/delete activity | Completed |
| Ullas | Added new fragment for managing course. (ManageCourseFragment) | Completed |
| Connected all of my modules and fragments. (ManageCourse, AddCourse, CourseListFragment) | Completed |
| Implemented Call back functions and Bundle arguments to send and receive data among fragments and Activity | Completed |
| Changed “AddCourseFragment” fragment to add as well as modify existing data | Completed |
| Made layout changes to add integration with the new fragment | Completed |
|  | Integrated Course Activity with Mohammad’s Student List Activity | Completed |
|  | Integrated Course Activity with Sowmya’s Module List Activity | Completed |
|  | Tested flow from Course Activity to Student List activity | Completed |
|  | Tested flow from Course Activity to Module List activity | Completed |
| Aditya | Designed layout for broadcast notification fragment and generate student groups fragment. | Completed |
| Implemented Lexicographic algorithm to generate the student groups. | Completed |
| Added junit test class to verify the correctness of the formation of the student groups. | Completed |
| Fetched the student list from the firebase database. | Completed |
| Implemented “GenerateStudentGroupsFragment” Fragment. | Completed |
| Implemented broadcast notification feature. | Completed |
| Added utility class to check null references. | Completed |
| Implemented Comparator to sort the student list according to last name. | Completed |
| Mohammed | Designed the firebase database structure for the ‘students’ root module and nested the modules, and courses in the ‘students’ database | Completed |
| Designed the layouts for student list per course, modules for a student per course, module items for every module item selected (e.g: assignments, in-class assignments, project etc), splits for a given module item. | Completed |
| Fetched the students list from the firebase database and displayed in a list | Completed |
| Fetched the selected students modules, selected module item (assignment, project etc), and splits for each module item and displayed the list in the respective layouts | Completed |
| Fetched the grades for all the modules and their splits and displayed in the selected module which is selected by the user for a particular student. | Completed |
| Implemented the fragments for student list, modules list per student, module item list for selected module and the splits with all the scores for all the selected item. | Completed |
| Implemented activity transition from student list to students module via fragments and passing the intent data to other activity for student and module reference | Completed |
| Implemented activity transition from module list to selected module item with the grades for that module item and whole list of the module items per module. | Completed |
| Implemented activity transition from module item to item split with grades for all the splits using fragments and intent data. | Completed |
| Implemented search functionality in the student list view | In Progress |

* 1. Phase 2 - Week 3

|  |  |  |
| --- | --- | --- |
| Sowmya | Came up with the UX design for the application. | Completed |
| Incorporated the UX look into “Module List”, “Sub Module List”, “Add Assignments”, “Add Modules”, “Edit Delete Module”. | Completed |
| Integrated transition from sub modules to grading students screen. | Completed |
| Made firebase changes to store module structure based on courseid. | Completed |
| Implemented an intent Services for updating the module details(add, edit and delete module) under every student for that course. | Completed |
| Implemented an intent Services for updating the submodule(add, edit and delete sub module) details under every student for that course. | Completed |
| Maintained the list of modules and sub modules mapping to handle scenarios where the module name is edited after a sub module has been added to it. | Completed |
| Created 9-patch images to be used in “Module List” and “Sub Module” list screen.. | Completed |
| Implemented drawer layout for 5 screens(“Module List”, “Sub Module List”, “Add Assignments”, “Add Modules”, “Edit Delete Module”) for easy navigations and sign out functionality. | Completed |
| Implemented slide animation for transition from one screen to another. (For 5 screens this has been done,“Module List”, “Sub Module List”, “Add Assignments”, “Add Modules”, “Edit Delete Module”). | Completed |
|  | Implemented showcase to show tips for users on operating the app, on the first installation of the app. (This has been done for 5 screens,“Module List”, “Sub Module List”, “Add Assignments”, “Add Modules”, “Edit Delete Module”). | Completed |
|  | Added progress “loading icon” which is displayed when data is still being fetched from Firebase, this icon disappears when data is loaded. (In “Module List”, “Sub Module List”). | Completed |
|  | Used “preference framework” for taking user information on whether the user wants to enable/disable notifications. | Completed |
|  | On login, used system service to check if “Internet” is available. If it is not available redirected user to mobile settings for enabling data. | Completed |
|  | Added few test cases. | Completed |
| Ullas | Added a new fragment to add TAs to the course list | Completed |
|  | Implemented Drop down and Custom array adapter to handle TA additions | Completed |
|  | Implemented feature that allows the respective courses to be displayed that the TA is part of | Completed |
|  | Implemented Shared Preferences to keep track of the user who is using the app and to display his/her courses | Completed |
|  | Made changes to the method that imports student list | Completed |
|  | Implemented permission checking and permission granting for marshmallow devices for file import | Completed |
|  | Made layout changes to add\_course\_fragment and course\_list\_fragment | Completed |
|  | If a TA is creating the course, add his name to the TA list automatically | Completed |
|  | Added error checking to check empty values before sending to firebase for add\_course\_fragment | Completed |
|  | Fixed a bug where all the courses were getting deleted sometimes | Completed |
|  | Fixed a bug which causes duplicate courses to be displayed | Completed |
|  | Added test cases for Import student function | Completed |
| Aditya | Implemented generic toolbar and added logout functionality | Completed |
| Implemented Side Navigation Bar and displayed profile information such as email, name and photo | Completed |
| Fetch the TAs pertaining to courses from the firebase | Completed |
| Implemented generic list adapter | Completed |
| Fetch the students from a course from the firebase. | Completed |
| Integrated student group fragment with mange course activity. | Completed |
| Integrate broadcast Email fragment with mange course activity. | Completed |
| Used 9 patch to create images. | Completed |
| Implemented asynctask to fetch the google profile pic | Completed |
| Implemented cloud push messaging via firebase console | Completed |
| Integrated List View with the UI for generate student groups fragment | Completed |
| Mohammed | Implemented the grading functionality for a selected course, module and module item | Completed |
|  | Implemented Activity, Fragments and Adapter for Grading without splits for a module item for a student | Completed |
|  | Implemented Activity, Fragments and Adapter for Grading with splits for a module item for a student | Completed |
|  | Implemented the Grading of each and every split for a module item if present and calculate total grade for that. | Completed |
|  | Grading for module item which does not have splits and storing it in the database under the graded student | Completed |
|  | Screen transition from Module Item to Student List for that course and Implemented search functionality in student list | Completed |
|  | Implemented Layout files for Grading with splits and without splits and integrated with activity and fragments | Completed |
|  | Updating grades for module item with splits and without splits for the selected course. | Completed |
|  | Tested the app and designed all the major test cases for the app | Completed |
|  | Test case designing done with pre-conditions, post-conditions, exceptions, steps, expected output and results for the test cases | Completed |

1. Test Cases

Test Case: 1 | Login

|  |  |
| --- | --- |
| **Test Case #:** 1  **System:** TA-Helper  **Designed by:** Mohammed Shethwala | **Test Case Name:** Login  **Short Description:** Login to the TA-Helper app with a google account |

|  |
| --- |
| **Pre-conditions:** User must have a valid google account. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Action** | **Expected System Response** | **Actual Result** |
| 1 | Install TA-Helper App | App Installed successfully | As Expected |
| 2 | Click on the Sign-In button | A popup showing the google accounts already logged in to the android phone. | As Expected |
| 3 | When prompted for email-id, select from the list or add a new email-Id | Sign In successful and Course List activity is displayed | As Expected |

|  |
| --- |
| **Exceptions:** No Internet Connection. In such case, user will not be able to sign-in |

|  |
| --- |
| **Post-conditions:** User successfully logged into the app. |

***Test Case: 2 | Logout***

|  |  |
| --- | --- |
| **Test Case #:** 2  **System:** TA-Helper  **Designed by:** Mohammed Shethwala | **Test Case Name:** Logout  **Short Description:** Logout of the TA-Helper app |

|  |
| --- |
| **Pre-conditions:** User must be logged into the app. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Action** | **Expected System Response** | **Actual Result** |
| 1 | Swipe right from the left side of the phone screen | Drawer opened on the left with multiple options | As Expected |
| 2 | Click on the Sign-Out button | User signed out of the app. Login page displayed | As Expected |

|  |
| --- |
| **Exceptions:** No Exceptions |

|  |
| --- |
| **Post-conditions:** User successfully logged out of the app. |

***Test Case: 3 | Course List***

|  |  |
| --- | --- |
| **Test Case #:** 3  **System:** TA-Helper  **Designed by:** Mohammed Shethwala | **Test Case Name:** Course List  **Short Description:** View Course List |

|  |
| --- |
| **Pre-conditions:** Have a valid login id (google email id) |

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Action** | **Expected System Response** | **Actual Result** |
| 1 | Click Login | Display Course List with all the courses for the TA | As Expected |

|  |
| --- |
| **Exceptions:** No Internet |

|  |
| --- |
| **Post-conditions:** A course list is displayed to the user |

***Test Case: 4 | Add Course***

|  |  |
| --- | --- |
| **Test Case #:** 4  **System:** TA-Helper  **Designed by:** Mohammed Shethwala | **Test Case Name:** Add Course  **Short Description:** User can add a course to the existing course list |

|  |
| --- |
| **Pre-conditions:** User must be logged in to the app |

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Action** | **Expected System Response** | **Actual Result** |
| 1 | Click on Add Course button | Add Course page is displayed with the multiple textfields | As Expected |
| 2 | Enter Course name, Course ID, Professor First Name, Professor Last Name, Professor Username, TA’s email Id’s and Professor’s Email Id’s in the Text fields | App captures the text entered by the user and displays it in the text fields | As Expected |
| 3 | Click on Add Course button | New course added to the database and the Course List is also updated with the new course | As Expected |

|  |
| --- |
| **Exceptions:** No Internet |

|  |
| --- |
| **Post-conditions:** New course successfully added to the course list and also updated in the database. |

***Test Case: 5 | Edit Course***

|  |  |
| --- | --- |
| **Test Case #:** 5  **System:** TA-Helper  **Designed by:** Mohammed Shethwala | **Test Case Name:** Edit Course  **Short Description:** User can edit a course from the course list |

|  |
| --- |
| **Pre-conditions:** At least one course should be there in the course list to edit |

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Action** | **Expected System Response** | **Actual Result** |
| 1 | Click on the Edit Button on the course you want to edit | Edit Course page is displayed with all the course information displayed in the respective text fields | As Expected |
| 2 | Edit any of the text field you want to edit | Text field value gets changed to whatever the user types into that text field | As Expected |
| 3 | Click Edit | The new values gets replaced by the old values in the database for the course which was edited | As Expected |

|  |
| --- |
| **Exceptions:** No Internet |

|  |
| --- |
| **Post-conditions:** Course successfully updated in the database |

***Test Case: 6 | Manage Course List***

|  |  |
| --- | --- |
| **Test Case #:** 6  **System:** TA-Helper  **Designed by:** Mohammed Shethwala | **Test Case Name:** Manage Course List  **Short Description:** User can view the Manage Course List |

|  |
| --- |
| **Pre-conditions:** At least one course should be available in the course list |

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Action** | **Expected System Response** | **Actual Result** |
| 1 | Click on Manage button from the course you want to manage | App displays the Manage Course List which has the following rows:  1.Import Student List  2.Student List  3.Send Broadcast Email  4.Export Records  5.Generate Student Groups | As Expected |

|  |
| --- |
| **Exceptions:**  No Internet |

|  |
| --- |
| **Post-conditions:** User can view the Manage Course List |

***Test Case: 7 | Import Student List***

|  |  |
| --- | --- |
| **Test Case #:** 7  **System:** TA-Helper  **Designed by:** Ullas Aithal Kota | **Test Case Name:** Import Student List  **Short Description:** User can import the student list from the excel file sheet and upload it to the database |

|  |
| --- |
| **Pre-conditions:** Excel file should be available in the Android Internal Storage/Download folder |

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Action** | **Expected System Response** | **Actual Result** |
| 1 | Click on the Manage button from any course you want to import the student list | Manage Course List is displayed to the user | As Expected |
| 2 | Click on Import Student List | Data from excel sheet is read and students are added in the database for the selected course | As Expected |
| 3 | Select the file from the file manager | File selected and back to the app | As expected |
| 4 | Permission pop up window - Clicked on Allow | Import successful | As expected |

|  |
| --- |
| **Exceptions:** No Internet |

|  |
| --- |
| **Post-conditions:** Students are added in the database under the course from which the import was done |

***Test Case: 8 | View Student List***

|  |  |
| --- | --- |
| **Test Case #:** 8  **System:** TA-Helper  **Designed by:** Mohammed Shethwala | **Test Case Name:** View Student List  **Short Description:** Student List for the selected course is displayed |

|  |
| --- |
| **Pre-conditions:** User must be logged in |

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Action** | **Expected System Response** | **Actual Result** |
| 1 | Click on the Manage button from any course you want to import the student list | Manage Course List is displayed to the user | As Expected |
| 2 | Click on Student List | App will display the student list for the course on which Manage button was clicked in the course list page | As Expected |

|  |
| --- |
| **Exceptions:** Students not imported from the excel sheet into the database |

|  |
| --- |
| **Post-conditions:** User can view the list of students for the selected course |

***Test Case: 9 Send Broadcast Email***

|  |  |
| --- | --- |
| **Test Case #:** 9  **System:** TA-Helper  **Designed by:** Aditya Jadhav | **Test Case Name:** Send Broadcast Email  **Short Description:** User will able to send broadcast email to students in the course |

|  |
| --- |
| **Pre-conditions:** 1. User must be logged in  2. Students must be present in the course |

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Action** | **Expected System Response** | **Actual Result** |
| 1 | Navigate to Send Broadcast Email through manage button from any course | Send Broadcast Email Screen is displayed to the user | As Expected |
| 2 | User compose a mail and click on send button | List the email clients through which user want to send email | As Expected |
| 3 | User will choose the client through which email will send | Open the email client and fill the field specify by user | As Expected |

|  |
| --- |
| **Exceptions:** If users does not have any email client on the mobile |

|  |
| --- |
| **Post-conditions:** Email sent to all the student in the course |

***Test Case: 10 Export Records***

|  |  |
| --- | --- |
| **Test Case #:**  **System:** TA-Helper  **Designed by:** Mohammed Shethwala | **Test Case Name:** Export Records  **Short Description:** |

|  |
| --- |
| **Pre-conditions:** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Action** | **Expected System Response** | **Actual Result** |
|  |  |  | As Expected |

|  |
| --- |
| **Exceptions:** |

|  |
| --- |
| **Post-conditions:** |

***Test Case: 11 Generate Student Groups***

|  |  |
| --- | --- |
| **Test Case #:**  **System:** TA-Helper  **Designed by:** Aditya Jadhav | **Test Case Name:** Generate Student Groups  **Short Description:** User is able to generate groups from the student list by specifying the group size |

|  |
| --- |
| **Pre-conditions:** 1. User must be logged in  2. Students must be present in the course |

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Action** | **Expected System Response** | **Actual Result** |
| 1 | Navigate to generate student groups through manage button from any course | Generate Student Groups Screen is displayed to the user | As Expected |
| 2 | User enters group size then press generate | User will able to see the students list with their group name | As Expected |

|  |
| --- |
| **Exceptions:** If the group size is not specified or the group size is not valid (less than or equal to 0) then toast message will be shown asking user to specify the size. |
| **Post-conditions:** Student Groups created and displayed |

***Test Case: 12 Search for a Student***

|  |  |
| --- | --- |
| **Test Case #:** 12  **System:** TA-Helper  **Designed by:** Mohammed Shethwala | **Test Case Name:** Search for a student  **Short Description:** User can search for a student from the list of students |

|  |
| --- |
| **Pre-conditions:** User must be in the Student List screen |

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Action** | **Expected System Response** | **Actual Result** |
| 1 | Click on search bar | App displays a keyboard to the user | As Expected |
| 2 | Enter username which you want to search | App searches for students and displays the searched student or multiple students based on the search string entered | As Expected |

|  |
| --- |
| **Exceptions:** Student List is empty |

|  |
| --- |
| **Post-conditions:** Searched student is displayed in the student list |

***Test Case: 13 View Student Modules***

|  |  |
| --- | --- |
| **Test Case #:** 13  **System:** TA-Helper  **Designed by:** Mohammed Shethwala | **Test Case Name:** View Student Modules  **Short Description:** User can view the student modules for the selected course |

|  |
| --- |
| **Pre-conditions:** At least one module should exist in the database for a student under the selected course |

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Action** | **Expected System Response** | **Actual Result** |
| 1 | Click on Manage button from the course you want to view the student modules for | App displays the Manage Course screen | As Expected |
| 2 | Click on Student List from the displayed screen | App displays the student list for the selected course | As Expected |
| 3 | Click on the student you want to view the modules for | App displays the modules for the selected student for the selected course | As Expected |

|  |
| --- |
| **Exceptions:** No Internet |

|  |
| --- |
| **Post-conditions:** User can view the modules for each and every student per every course |

***Test Case: 14 | View Module Item Score With Splits***

|  |  |
| --- | --- |
| **Test Case #:** 14  **System:** TA-Helper  **Designed by:** Mohammed Shethwala | **Test Case Name:** View Module Item Score for a student with splits  **Short Description:** ” |

|  |
| --- |
| **Pre-conditions:** Module Item with split should exist for the selected course |

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Action** | **Expected System Response** | **Actual Result** |
| 1 | Click on Manage button from the course you want to view the student modules for | App displays the Manage Course screen | As Expected |
| 2 | Click on Student List from the displayed screen | App displays the student list for the selected course | As Expected |
| 3 | Click on the student you want to view the modules for | App displays the modules for the selected student for the selected course | As Expected |
| 4 | Click on the Module for which you want to view the Module Items for that student | App displays the list of module items in the selected module for the selected student with the gained marks and maximum possible marks for that module item | As Expected |
| 5 | Click on a Module Item from the list which has splits | App displays the list of splits for the selected Module Item and also displays the gained marks and maximum possible marks for that split for the selected student | As Expected |

|  |
| --- |
| **Exceptions:** No Internet |

|  |
| --- |
| **Post-conditions:** Marks for selected student for the selected module item with splits is displayed to the user |

***Test Case: 15 | View Module Item score without Splits***

|  |  |
| --- | --- |
| **Test Case #:** 15  **System:** TA-Helper  **Designed by:** Mohammed Shethwala | **Test Case Name:** View Module Item Score for a student without splits  **Short Description:** ” |

|  |
| --- |
| **Pre-conditions:** Module Item without split should exist for the selected course |

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Action** | **Expected System Response** | **Actual Result** |
| 1 | Click on Manage button from the course you want to view the student modules for | App displays the Manage Course screen | As Expected |
| 2 | Click on Student List from the displayed screen | App displays the student list for the selected course | As Expected |
| 3 | Click on the student you want to view the modules for | App displays the modules for the selected student for the selected course | As Expected |
| 4 | Click on the Module for which you want to view the Module Items for that student | App displays the list of module items in the selected module for the selected student with the gained marks and maximum possible marks for that module item | As Expected |
| 5 | Click on the Module Item | App displays that there are no splits for the selected Module Item | As Expected |

|  |
| --- |
| **Exceptions:** No Internet |

|  |
| --- |
| **Post-conditions:** Marks for selected student for the selected module item without splits is displayed to the user |

***Test Case: 16 | Add New Module***

|  |  |
| --- | --- |
| **Test Case #:** 16  **System:** TA-Helper  **Designed by:** Mohammed Shethwala | **Test Case Name:** Add New Module  **Short Description:** User can add a new module to the selected course |

|  |
| --- |
| **Pre-conditions:** A Course must be available in which the new module can be added |

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Action** | **Expected System Response** | **Actual Result** |
| 1 | Click Grade button on any one course from the list of courses | Module List for the selected course is displayed to the user | As Expected |
| 2 | Click on Add New Module | Add New Module screen is displayed | As Expected |
| 3 | Enter Module Name | Text field displays the entered text by the user | As Expected |
| 4 | Click Add Module | Module is added to the database under the selected course and updated module list with the newly added module is displayed | As Expected |

|  |
| --- |
| **Exceptions:** No Internet |

|  |
| --- |
| **Post-conditions:** New Module updated in the Module List and added to the database under the selected course |

***Test Case: 17 | Add Module Item with Splits***

|  |  |
| --- | --- |
| **Test Case #:** 17  **System:** TA-Helper  **Designed by:** Mohammed Shethwala | **Test Case Name:** Add Module Item with Splits  **Short Description:** User can add a new module item to a Module under a selected course which has splits |

|  |
| --- |
| **Pre-conditions:** A module under a selected course must exist. User must be in the Module List screen under Grade section |

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Action** | **Expected System Response** | **Actual Result** |
| 1 | Click on the View Module Items (Blue button on the right) | Module Items for the selected module is displayed | As Expected |
| 2 | Click Add Project Sub Module button on the top of the screen | Add Sub Module screen is displayed | As Expected |
| 3 | Enter Assignment Name in the first text field | Module Item name is displayed in the text field | As Expected |
| 4 | Enter Split Name and Split Score and click Add Split.  Repeat this step for n number of splits | Split Name and Split Score is fetched and added to the list of splits for this Module Item. Split List below is updated.  Repeated for n number of splits | As Expected |
| 5 | Click on Add Sub Module button in the bottom of the screen | Module Item along with the splits is stored in the database under the selected course.  Updated Module Item list with the new Module Item added is displayed | As Expected |

|  |
| --- |
| **Exceptions:** No Internet |

|  |
| --- |
| **Post-conditions:** Module Item added in the database under the selected course |

***Test Case: 18 | Add Module Item without Splits***

|  |  |
| --- | --- |
| **Test Case #:** 18  **System:** TA-Helper  **Designed by:** Mohammed Shethwala | **Test Case Name:** Add Module Item without splits  **Short Description:** User can add a new module item to a Module under a selected course which does not have splits |

|  |
| --- |
| **Pre-conditions:** A module under a selected course must exist. User must be in the Module List screen under Grade section |

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Action** | **Expected System Response** | **Actual Result** |
| 1 | Click on the View Module Items (Blue button on the right) | Module Items for the selected module is displayed | As Expected |
| 2 | Click Add Project Sub Module button on the top of the screen | Add Sub Module screen is displayed | As Expected |
| 3 | Enter Assignment Name in the first text field | Module Item name is displayed in the text field | As Expected |
| 4 | Click on Add Sub Module button | Module Item is stored in the database under the selected course.  Updated Module Item list with the new Module Item added is displayed | As Expected |

|  |
| --- |
| **Exceptions:** No Internet |

|  |
| --- |
| **Post-conditions:** Module Item added in the database under the selected course |

***Test Case: 19 | Edit Module Item with Splits***

|  |  |
| --- | --- |
| **Test Case #:** 19  **System:** TA-Helper  **Designed by:** Mohammed Shethwala | **Test Case Name:** Edit Module Item with Splits  **Short Description:** User can edit a Module Item which has splits |

|  |
| --- |
| **Pre-conditions:** A Module Item with splits should exist under the selected Module under the selected course |

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Action** | **Expected System Response** | **Actual Result** |
| 1 | Click on the Edit Module Button for the Module Item which has splits (Gray button with pencil on the left side) | Edit Module Item with the Module Item name, score and splits is displayed | As Expected |
| 2 | 1.Edit the Module Item Name  2.Add a new Split to the Split List  3.Delete a previous split from the Split List  4. Edit the Total  5. Click Delete button | 1. Module Item name gets changed in the text field  2. New Split is added and displayed in the Split List  3. Deleted Split gets removed from the Split List  4. Total gets changed in the Total text field  5. Step 3 will not be executed and the Module Item will be deleted from the Module along with all the splits and updated in the database | As Expected |
| 3 | Click on Edit button | App saves all the edited changes to the database and displays the Module List to the user | As Expected |

|  |
| --- |
| **Exceptions:** No Internet |

|  |
| --- |
| **Post-conditions:** Module Item with splits updated in the database |

***Test Case: 20 | Edit Module Items without Splits***

|  |  |
| --- | --- |
| **Test Case #:** 20  **System:** TA-Helper  **Designed by:** Mohammed Shethwala | **Test Case Name:** Edit Module Items without Splits  **Short Description:** User can edit Module Item which does not have a split and can add new splits to it |

|  |
| --- |
| **Pre-conditions:** A Module Item without splits should exist under the selected Module under the selected course |

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Action** | **Expected System Response** | **Actual Result** |
| 1 | Click on the Edit Module button for the Module Item which does not have splits (Gray button with pencil on the left side) | Edit Module Item with the Module Item name, score | As Expected |
| 2 | 1.Edit the Module Item Name  3.Add a new Split to the Split List  4. Edit the Total  4. Click Delete button | 1. Module Item name gets changed in the text field  2. New Split is added and displayed in the Split List  3. Total gets changed in the Total text field  4. Step 3 will not be executed and the Module Item will be deleted from the Module along with all the splits and updated in the database |  |

|  |
| --- |
| **Exceptions:** No Internet |

|  |
| --- |
| **Post-conditions:** Module Item updated in the database with added splits if any |

***Test Case: 21 Grade with Splits***

|  |  |
| --- | --- |
| **Test Case #:** 21  **System:** TA-Helper  **Designed by:** Mohammed Shethwala | **Test Case Name:** Grade (Edit Grade) with Splits  **Short Description:** User can grade a student for a Module Item which has splits |

|  |
| --- |
| **Pre-conditions:** Module Item with splits for the selected course should exist. User must be in the Module Item List under selected Module under selected Course |

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Action** | **Expected System Response** | **Actual Result** |
| 1 | Click on the Grade Module Item button for the Module Item which has splits (Blue button on the right side) | Student List for that selected course is displayed | As Expected |
| 2 | Select a student from the Student List to grade | Grade With Splits screen is displayed | As Expected |
| 3 | Grade/Edit the Split scores from the Split List (Enter numbers in the grading text fields) | Changed value for split is displayed in the changed text field | As Expected |
| 4 | Click Calculate Button | 1. If any of the score for splits entered is more than maximum possible marks, then user will be notified to change the marks below the threshold.  2. Else, score is calculated from the splits and displayed in the Total text field | As Expected |
| 5 | Click Submit Button | Marks from the splits as well as the total marks are saved in the database for the selected student. | As Expected |

|  |
| --- |
| **Exceptions:** No Internet |

|  |
| --- |
| **Post-conditions:** New Grades updated for the student selected |

***Test Case: 22 | Grade without Splits***

|  |  |
| --- | --- |
| **Test Case #:** 22  **System:** TA-Helper  **Designed by:** Mohammed Shethwala | **Test Case Name:** Grade (Edit Grade) without Splits  **Short Description:** User can grade a student for a Module Item which does not have a split |

|  |
| --- |
| **Pre-conditions:** Module Item without splits for the selected course should exist. User must be in the Module Item List under selected Module under selected Course |

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Action** | **Expected System Response** | **Actual Result** |
| 1 | Click on the Grade Module Item button for the Module Item which does not have splits (Blue button on the right side) | Student List for that selected course is displayed | As Expected |
| 2 | Select a student from the Student List to grade | Grade Without Splits screen is displayed | As Expected |
| 3 | Enter/Edit the marks in the Total Score text field | Entered/Changed score is displayed | As Expected |
| 4 | Click Submit | Graded score for the student is saved/updated in the database | As Expected |

|  |
| --- |
| **Exceptions:** No Internet |

|  |
| --- |
| **Post-conditions:** New Grades updated for the student selected |

***Test Case: 23 | Edit Module***

|  |  |
| --- | --- |
| **Test Case #:** 23  **System:** TA-Helper  **Designed by:** Mohammed Shethwala | **Test Case Name:** Edit Module  **Short Description:** User can Edit the name of Module or delete a module |

|  |
| --- |
| **Pre-conditions:** A module under the selected course must exist. User must be in the Module List screen under the Grade Course section |

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Action** | **Expected System Response** | **Actual Result** |
| 1 | Click on the Edit Module button for the Module  which you want to edit (Gray button on the left side with pencil icon) | Edit Module screen is displayed | As Expected |
| 2 | 1.Edit the Module name and click Edit.  2. Click Delete | 1. Edited Module Name is updated in the database wherever it is referenced.  2. The Module gets deleted from the database along with all the Module Items for this Module. | As Expected |

|  |
| --- |
| **Exceptions:** No Internet |

|  |
| --- |
| **Post-conditions:** Module name edited or Module delete from the database |

***Test Case: 24 | Enable/Disable Notifications***

|  |  |
| --- | --- |
| **Test Case #:** 24  **System:** TA-Helper  **Designed by:** Mohammed Shethwala | **Test Case Name:** Enable/Disable Notification  **Short Description:** User can enable/disable notifications for the app |

|  |
| --- |
| **Pre-conditions:** User must be logged into the the app |

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Action** | **Expected System Response** | **Actual Result** |
| 1 | Swipe right from the left edge of the screen | Drawer with multiple options is displayed | As Expected |
| 2 | Click Settings | Settings page for the app is displayed | As Expected |
| 3 | Click the checkbox either to enable or disable notifications | App gets the checkbox click and updates the notification setting for the user | As Expected |

|  |
| --- |
| **Exceptions:** No Internet |

|  |
| --- |
| **Post-Conditions:** Notification successfully enabled/disabled. |

***Test Case: 25 | Edit Module without changing name***

|  |  |
| --- | --- |
| **Test Case #:** 25  **System:** TA-Helper  **Designed by:** Sowmya Parameshwara | **Test Case Name:** Edit Module without changing name  **Short Description:** User can go to edit screen and click edit without changing module name. |

|  |
| --- |
| **Pre-conditions:** User must have created the module. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Action** | **Expected System Response** | **Actual Result** |
| 1 | Click on edit from the module list screen (pencil button on left side). | Edit module screen must be launched. | As Expected |
| 2 | Do not edit the module name and click edit button. | The module and sub modules corresponding to the items should remain unaltered in firebase under module and students structure. | As Expected |

|  |
| --- |
| **Exceptions:** No Internet |

|  |
| --- |
| **Post-conditions:** Module details remain unedited. |

***Test Case: 26 | Edit Module name for a module which as sub modules defined.***

|  |  |
| --- | --- |
| **Test Case #:** 26  **System:** TA-Helper  **Designed by:** Sowmya Parameshwara | **Test Case Name:** Edit Module name for a module which as sub modules defined.  **Short Description:** User can edit the module name for a module which has sub modules defined, the edited module should retain the existing sub module. |

|  |
| --- |
| **Pre-conditions:** User must have created a module. This module must have one sub module without split and one sub module with split. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Action** | **Expected System Response** | **Actual Result** |
| 1 | Click on edit from the module list screen (pencil button on left side). | Edit module screen must be launched. | As Expected |
| 2 | Edit the module name and click edit button. | The old module name must be replaced with new module name entered. Under this new module name , the existing sub module (one sub module with split and one without split) must be displayed both under module structure and under student structure. | As Expected |

|  |
| --- |
| **Exceptions:** No Internet |

|  |
| --- |
| **Post-conditions:** Module name gets edited, without losing sub module information. |

***Test Case: 27 | Delete Module.***

|  |  |
| --- | --- |
| **Test Case #:** 27  **System:** TA-Helper  **Designed by:** Sowmya Parameshwara | **Test Case Name:** Delete an existing module.  **Short Description:** User can delete a module and it’s details along with its sub modules will get removed from the system. |

|  |
| --- |
| **Pre-conditions:** User must have created a module. This module must have one sub module without split and one sub module with split. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Action** | **Expected System Response** | **Actual Result** |
| 1 | Click on edit from the module list screen (pencil button on left side). | Edit module screen must be launched. | As Expected |
| 2 | Click delete button. | The module along with it’s sub module must be removed from module and student structure. | As Expected |

|  |
| --- |
| **Exceptions:** No Internet |

|  |
| --- |
| **Post-conditions:** Module gets deleted. |

***Test Case: 28 | TA managing multiple courses***

|  |  |
| --- | --- |
| **Test Case #:** 28  **System:** TA-Helper  **Designed by:** Sowmya Parameshwara | **Test Case Name:** TA managing multiple courses.  **Short Description:** Same user can manage multiple classes and the class data should not be mixed up. |

|  |
| --- |
| **Pre-conditions:** User must have a course “CourseOne” under their account. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Action** | **Expected System Response** | **Actual Result** |
| 1 | User logs in and goes to course list screen. | Course list screen must be launched with a single course. | As Expected |
| 2 | Click add button to add a new course. | A new course should get added. | As Expected |
| 3 | Click “Import Student” and import student list for this new course. (Have few students who are part of CourseOne and part of this new course.) | Student list should get imported. | As Expected |
| 4 | Click “Student list” for this newly added course. | The student details should be correct. | As Expected |
| 5 | Go back to “CourseOne”, click “Manage”, click “Student List”. | The student details for “CourseOne” should be unchanged. | As Expected |

|  |
| --- |
| **Exceptions:** No Internet |

|  |
| --- |
| **Post-conditions:** TA is able to successfully manage multiple courses.. |

***Test Case: 29 | Multiple TA’s collaborating on a single course***

|  |  |
| --- | --- |
| **Test Case #:** 29  **System:** TA-Helper  **Designed by:** Sowmya Parameshwara | **Test Case Name:** Multiple TA’s collaborating on a single course.  **Short Description:** Multiple TA’s should be able to collaborate on a single course. |

|  |
| --- |
| **Pre-conditions:** A course must be created with more than one TA for it.. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Action** | **Expected System Response** | **Actual Result** |
| 1 | User creates a course and adds more than one TA for that course. | Course list screen must be launched with that course added to the list. | As Expected |
| 2 | The other TA logs into his account. | He must be able to see the newly added course in his course list. | As Expected |
| 3 | TA1 and TA3 adds few modules and sub modules to that course. | Both TA’s should be able to see all the modules and sub modules which were added. | As Expected |
| 4 | TA1 and TA2 grades students. | The updated scores should reflect in both their account. | As Expected |

|  |
| --- |
| **Exceptions:** No Internet |

|  |
| --- |
| **Post-conditions:** They should be successfully able to collaborate. |

***Test Case: 30 | Import Student List***

|  |  |
| --- | --- |
| **Test Case #:** 30  **System:** TA-Helper  **Designed by:** Ullas Aithal Kota | **Test Case Name:** Import Student List  **Short Description:** User can import the student list from the excel file sheet and upload it to the database |

|  |
| --- |
| **Pre-conditions:** Excel file should be available in the Android Internal Storage/Download folder |

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Action** | **Expected System Response** | **Actual Result** |
| 1 | Click on the Manage button from any course you want to import the student list | Manage Course List is displayed to the user | As Expected |
| 2 | Click on Import Student List | Data from excel sheet is read and students are added in the database for the selected course | As Expected |
| 3 | Select the file from the file manager | File selected and back to the app | As expected |
| 4 | Permission pop up window - Clicked on Deny | Import failed. A toast is displayed with the error | As expected |

|  |
| --- |
| **Exceptions:** No Internet |

|  |
| --- |
| **Post-conditions:** Students are added in the database under the course from which the import was done |

***Test Case: 31 | Import Student List***

|  |  |
| --- | --- |
| **Test Case #:** 31  **System:** TA-Helper  **Designed by:** Ullas Aithal Kota | **Test Case Name:** Import Student List  **Short Description:** User can import the student list from the excel file sheet and upload it to the database |

|  |
| --- |
| **Pre-conditions:** Excel file should be available in the Android Internal Storage/Download folder |

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Action** | **Expected System Response** | **Actual Result** |
| 1 | Click on the Manage button from any course you want to import the student list | Manage Course List is displayed to the user | As Expected |
| 2 | Click on Import Student List | Data from excel sheet is read and students are added in the database for the selected course | As Expected |
| 3 | Select the file from the file manager | File selected and back to the app | As expected |
| 4 | Permission pop up window - Clicked on Allow | Import successful | As expected |
| 5 | Selected file not excel file | A toast message is displayed stating the file is not excel | As expected |

|  |
| --- |
| **Exceptions:** No Internet |

|  |
| --- |
| **Post-conditions:** Students are added in the database under the course from which the import was done |

***Test Case: 32 | Import Student List***

|  |  |
| --- | --- |
| **Test Case #:** 32  **System:** TA-Helper  **Designed by:** Ullas Aithal Kota | **Test Case Name:** Export Assignment Score List  **Short Description:** The assignment score can be exported as excel |

|  |
| --- |
| **Pre-conditions:** At least 1 assignment has to be graded |

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Action** | **Expected System Response** | **Actual Result** |
| 1 | Click on the Manage button from any course you want to import the student list | Manage Course List is displayed to the user | As Expected |
| 2 | Click on Export Records button | A new window with the list of assignments and thesis are shown | As Expected |
| 3 | Select any one of the assignment to export from the drop down | An assignment is selected | As expected |
| 4 | Click on Export Button | The file will be exported to internal storage of the app | As expected |

|  |
| --- |
| **Exceptions:** No Internet |

|  |
| --- |
| **Post-conditions:** A excel file with the filename as the assignment number will be saved in the internal storage of the device |

***Test Case: 33 | TA Selection List***

|  |  |
| --- | --- |
| **Test Case #:** 33  **System:** TA-Helper  **Designed by:** Ullas Aithal Kota | **Test Case Name:** TA Selection List  **Short Description:** Add the TAs to the course while adding a new course |

|  |
| --- |
| **Pre-conditions:** More than 2 users has to be logged in |

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Action** | **Expected System Response** | **Actual Result** |
| 1 | Click on the Add Course button after logging in | A new window with the required course details will appear. Enter all the required fields | As Expected |
| 2 | Click on Add TAs button | A new window with a drop down holding a list of TAs will appear | As Expected |
| 3 | Select a TA and click on Add TA | The added TA will now appear in the list below | As expected |
| 4 | Click on Add TAs | Previous window will reappear with the list of TAs that were added | As expected |

|  |
| --- |
| **Exceptions:** No Internet |

1. Instructions for testing the app:
2. Open the app
3. Sign in with your hawkid
4. Click on “Add Course” to add a course
5. Enter all the details
6. Select the TAs that you want to add to the course by clicking on “Add\Remove TAs” and selecting the TAs from drop down
7. Click on Add TAs
8. Click on “Add Course” to add the course
9. The course should now be visible in the list
10. Click on “Manage”

10) Click on “Import Students List” to import “Students.xls” from your phone

11) Allow the permissions for the app

1. Students list will be imported
2. Important Information
3. If you are building the app, you will notice a lot of grade build errors. This is because we are using APACHE’s POI API to parse the excel file and provide the result. This is a known bug in the API. **But nevertheless, the app will get installed all the functionalities will work as expected**
4. The “Students.xlsx” should have data in the following format
   1. Column 1 : Student username
   2. Column2 : Student email id
   3. Column 3: Student Last Name
   4. Column 4: Student First Name

3) If you’re building the app, then you might not able to login to the app because the build is tightly coupled with firebase certificate. So to run this app its better to get the apk and install it on the phone. The apk is attached in the zip.