Pathways Mobile Transfer Doc

Building a Mobile App for Students and Universities from the Ground Up

SCALE Learning Technologies

Mentors:

Tiffany Reiss, Osman Bakari, Jack Wharton, Jimmy Zheng, Christopher Nathman, Matthew Johnson, Ernest Spicer

Team LD:

Ganeshram Krishnamoorthy, Caleb Lee, Yihui Fang, Wenda Liu

Table of Contents

Tools Used

Tools/Libraries Used	
REACT Native	UI Software framework that allows a unified code base between Android and Apple development.
NEO4J	Graph database used to visualize student schedule/course requirements.
Android Studio	Mainly used for emulation of the app.
NPM	Package Management for Node JS.
EXPO GO	Testing REACT Native apps on Android and IOS devices.
POSTMAN	Backend testing. Routes for data to be transferred from backend to front end and visa versa.
Docker Desktop	Local Network server simulation.
Visual Studio Code	Main IDE, “Live Share” extension.

Works Completed

Sign-in Page:
Can authenticate the user and give feedback if sign-in information is incorrect.

Create Account page:
Can create a user account, however there are schemas for some inputs such as the password which are required by the back-end that needs to be considered such as password length.

University Page:
Displays university information from the back end

Degrees Page:
Display the degrees that the user can make a schedule for (we have been mainly testing Computer Science degrees as we know this is one of the degrees which is fully supported by the back-end we were supplied with.

Terms Page:
Display the Terms, however some terms may cause issues in schedule generation due to an issue with the current back-end.

Schedule Page:
Displays the schedule the user has created. As mentioned previously, it may display incorrectly due to a back-end bug (the data is being displayed exactly as it is received from the back-end).

Profile Page:
Display the user information, allow them to edit the user textual information and take the profile picture.

Work In-Progress and TODO

Degrees/Terms Page:
This page needs to fully connect to the backend since at the moment we are using a JSON file for the data.

Profile Page:
Currently we allow users to update their personal information, but the user is unable to upload an image as we face difficulty using the route.

Jest:
Began using this testing framework, however we were not able to get it working fully (it runs, however the test case we provided is not resulting as expected).

Incorporate Desktop Application Features:

Visualize Graph of Schedules:
Schedules can be visualized using a graphing library with connected nodes.

Editing Schedules:
Schedules should be able to be manually edited by the user when viewing the schedule.

Showing Course Details:
Include visually pleasing styling. Also, details of all courses should be able to be seen by the user when clicked on by the button.

System Components

Screens:

- CurrentSchedule

    - Displays the breakdown of the last schedule that the user selected.
- EditProfile
    - Used to edit the textual profile information such as phone number and state.
- ForgotPassword
    - Front-end for password reset
- Holder
    - Holds all the screens and contains the navigation bar on the bottom of the view.
- JoinDegree
    - Lets the user select their desired degree program
- JoinTerm
    - Lets the user select their desired term.
- LP_S_Both
    - Displays the names of all the generated schedules.
- LP_S_ Universities
    - Displays the Universities that the user could select.
- Profile
    - Displays the profile information of the user.
- ScheduleNaming
    - Used to name the schedule before generation.
- SigninScreen
    - Used to sign-in.
- SignupScreen
    - Used to create an account.
- index_*
    - Necessary in React Native for indexing and navigation

Most Important Components:

- CustomButton
    - Stylized button that allows us to call the component that has a base button function where all we need to do is give it styles.
- CustomInput
    - Stylized input field that any screen can use.
- Navbar-React
    - Library used to create the navigation bar for the app.
- Navigation
    - Use navigate.navigate(“screenName”) function in order to change screens.
    - Ensure that the screen index is implemented.
    - There is an “outer screen” AKA Holder that contains screens like Profile, Universities, etc. This outer screen holds the navbar and the upper title of each screen.

Bugs

- If a new user does not generate a schedule and immediately goes to the Schedule Page, the program will crash. The mitigation we attempted to utilize to prevent this from occurring is not working.
- The user may take and choose the profile picture, but issue with displaying the profile picture from the backend.