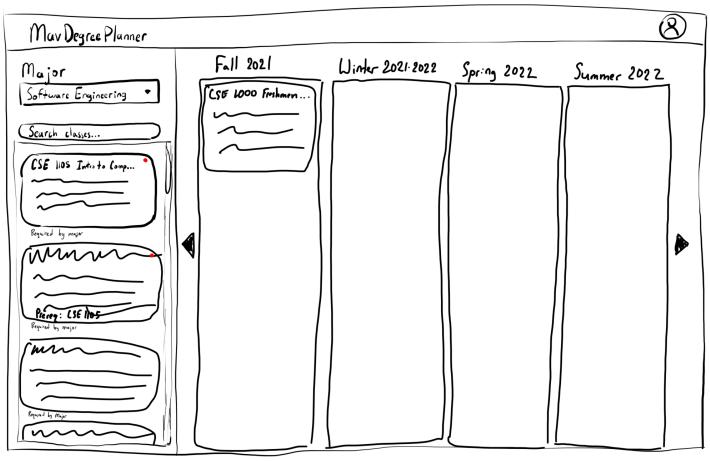
Department of Computer Science & Engineering
The University of Texas at Arlington

Details Design Specification
CSE 4317: Senior Design II
Spring 2022



Team IDP
Interactive Degree Planner

Daniel Newville
Le Uyen Nguyen
Ijaz Mohamed Umar
Safi Ullah

Revision History

Revision	Date	Author(s)	Description
0.1	02.24.2022	DN	document creation
0.2	02.24.2022	DN, IU, SU, LN	document draft
0.3	03.08.2022	DN, IU, SU, LN	document update
0.4	03.09.2022	DN, IU, SU, LN	document complete

CONTENTS

1 Introduction **5**
- 1.1 Purpose and Use . . . 5
- 1.2 Intended Audience . . . 5

2 System Overview **5**
- 2.1 Degree Planner UI Description . . . 6
- 2.2 Firebase Description . . . 6
- 2.3 Dashboard Description . . . 6

3 Degree Planner UI Layer **7**
- 3.1 Layer Hardware . . . 7
- 3.2 Layer Operating System . . . 7
- 3.3 Layer Software Dependencies . . . 7
- 3.4 Web Application . . . 7
- 3.5 Course Catalog . . . 8
- 3.6 SemesterCatalog . . . 9

4 Firebase Layer **10**
- 4.1 Layer Hardware . . . 10
- 4.2 Layer Operating System . . . 10
- 4.3 Layer Software Dependencies . . . 10
- 4.4 Authentication Subsystem . . . 10
- 4.5 Firestore Database Subsystem . . . 11

5 Dashboard Layer **12**
- 5.1 Layer Hardware . . . 12
- 5.2 Layer Operating System . . . 12
- 5.3 Layer Software Dependencies . . . 12
- 5.4 Homepage/Login . . . 12
- 5.5 Dashboard . . . 13

6 Appendix A **14**

List of Figures

1 IDP architectural layer diagram 5
2 Web Application Subsystem 7
3 CourseCatalog Subsystem 8
4 SemesterCatalog Subsystem 9
5 Diagram of the Firebase layer with the Authentication subsystem selected 10
6 Diagram of the Firebase layer with the Firestore Database subsystem selected 11
7 Homepage/Login Functionality 12
8 Dashboard Functionality 13

List of Tables

1 Introduction

Degree planning is never an easy task. It requires students to have a thorough understanding about the course information, description, pre-/co-requisites, and etc. Any mistake or carelessness during this process may result in wasted time, money, effort, and negative impact on goal accomplishment. Therefore, Interactive Degree Planner is developed with a mission to facilitate the degree planning process for students in the Department of Computer Science and Engineering of the University of Texas ar Arlington.

1.1 Purpose and Use

Interactive Degree Planner is a web-based application that is expected to simplify the process of planning an academic roadmap. The user should be able to log-in, choose the desired major in the CSE department, and start building his/her own degree planner. The website will abstract away all the complexity during this process, such as pre- and co-requisites, lookup, and etc. Besides, it takes advantage of drag-and-drop function in order to provide a user-friendly interface.

1.2 Intended Audience

The website aims to provide services for current and/or future CSE students at the University of Texas at Arlington.

2 System Overview

There are three layers. The layers are the UI layer, Firebase layer, and Dashboard layer. Each layer has interactions with the other layers, and they help build the overall system.

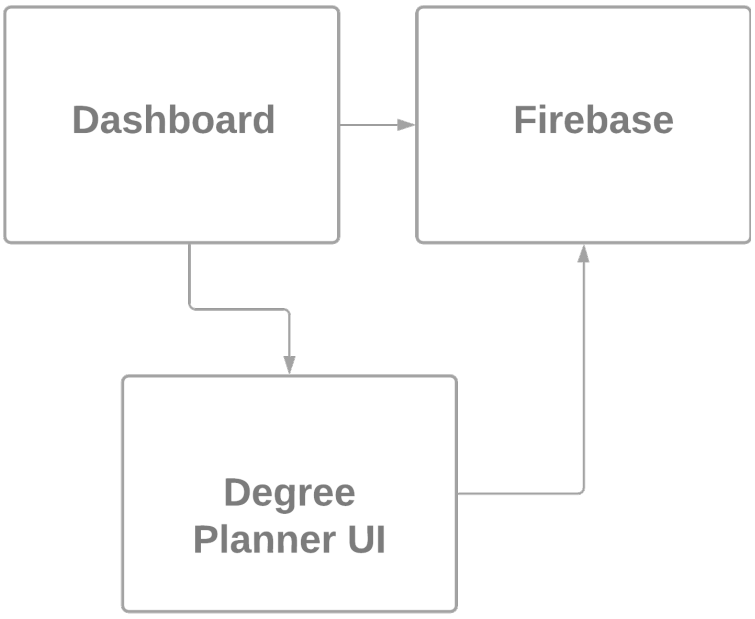


Figure 1: IDP architectural layer diagram

2.1 Degree Planner UI Description

The degree planner UI will have features allowing users to easily select and interact with their degree plans. It will allow users to select classes, display information, and give the user access to navigate between the Dashboard and the degree planning section. The UI will interact with Firebase when the student selects classes. It will get the data from Firebase and bring it to the UI layer. Once the UI has the data, it will then display the data to the users. The UI will also allow the users to navigate between screens and give them access to switch to the Dashboard screen. We will call this the Degree Planner UI layer.

2.2 Firebase Description

The firebase layer has features such as keeping data, authentication, and sending data. When a user selects classes the UI layer will request the data from firebase. Firebase will then send that data over to display in the UI layer. Firebase is where all the data is stored for classes and user accounts. It keeps the data organized and allows easier retreival. Firebase also works with the dashboard layer to authenticate accounts. When a user wants to make an account the users info will be sent over to firebase to be stored to authenticate when they sign in. Everytime the user signs in or out, firebase will authenticate that interaction. This layer will be called the firebase layer.

2.3 Dashboard Description

The dashboard layer of the degree planner will have a link to the degree planner, it will allow the user to sign in and out of their account, and it will let the user see the flowchart of their major. The user will be able to make an account in the hompage. They will give their information and be able to make an account. Once their account is made, their data will be sent over and stored in firebase. They can then sign in and out with the use of the dashboard and everytime they sign in and out. The authentication will be handled through firebase. In the dashboard there will also be the option for the user to view the flowchart of their major. There will also be a sidebar that shows some user information as well as links to other parts of the website. This layer will be called the dashboard layer.

3 Degree Planner UI Layer

The Degree Planner UI layer consists of Web application, Course catalog, and Semester catalog. It is responsible for processing/displaying information related to degree planning tasks and interacting with the users. Important to notice, only authenticated users can access this layer.

3.1 Layer Hardware

The Degree Planner UI layer uses the screen of any devices having web browser(s).

3.2 Layer Operating System

The Degree Planner UI does not require any specific operating system since it is a web-based application running on any web browser.

3.3 Layer Software Dependencies

The Degree Planner UI is a web-based application built with ReactJS. It is implemented with React libraries, including react-beautiful-dnd and styled-component. React Router takes responsibility for navigating.

3.4 Web Application

Web application subsystem provides the user interface for the process of planning degree. The two fundamental goals of this layer is to display information and to interact with user. All courses corresponding to the users' major will be displayed in this UI. The users are capable of dragging course items and drop them into any semester columns.

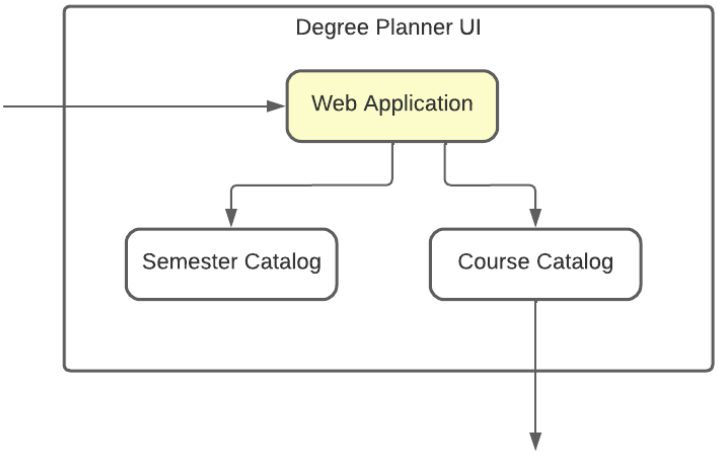


Figure 2: Web Application Subsystem

3.4.1 Subsystem Hardware

See section 3.1.

3.4.2 Subsystem Operating System

See section 3.2.

3.4.3 Subsystem Software Dependencies

See section 3.3.

3.4.4 Subsystem Programming Languages

This subsystem uses React with TypeScipt, and CSS.

3.4.5 Subsystem Data Structures

No particular data structure is required.

3.4.6 Subsystem Data Processing

Any data retrieved from Firestore database will be processed and formatted in Course and Semester subsystem. Web Application subsystem only receives the processed data from other subsystems and displays it to users.

3.5 Course Catalog

Course catalog subsystem fetches data from the database and sends it to the Web application subsystem. It is basically an interface between the Degree Planner UI layer and Firebase database layer.

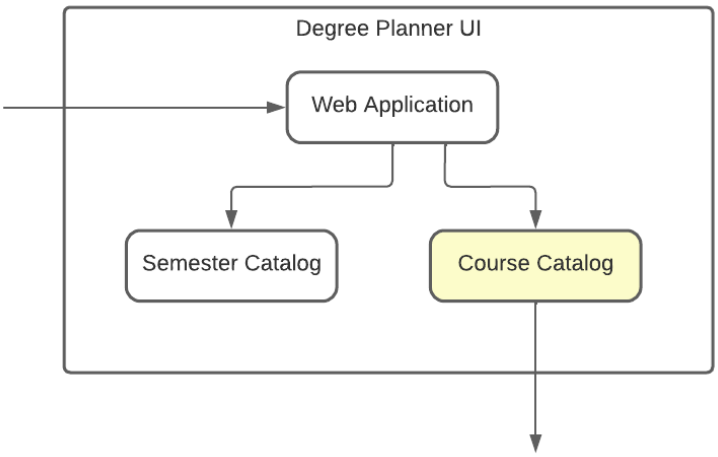


Figure 3: CourseCatalog Subsystem

3.5.1 Subsystem Hardware

See section 3.1.

3.5.2 Subsystem Operating System

See section 3.2.

3.5.3 Subsystem Software Dependencies

It will be implemented using the react-firebase library as well as Firestore database. For more information, see section 3.3.

3.5.4 Subsystem Programming Languages

This subsystem uses React with TypeScipt, and CSS.

3.5.5 Subsystem Data Structures

Each course retrieved from Firestore database into this subsystem has class Course. This user-defined object consists of nine properties, including document id, course id, course number, department, description, majors, availability, co-requisite, and pre-requisite courses.

3.5.6 Subsystem Data Processing

This subsystem will receive data from Firestore database and process the data. It will then give that data to the web application subsystem.

3.6 SemesterCatalog

Semester catalog subsystem reads data from the user information, and sends it to the Web application subsystem. It is basically an interface between UI layer and Dashboard layer.

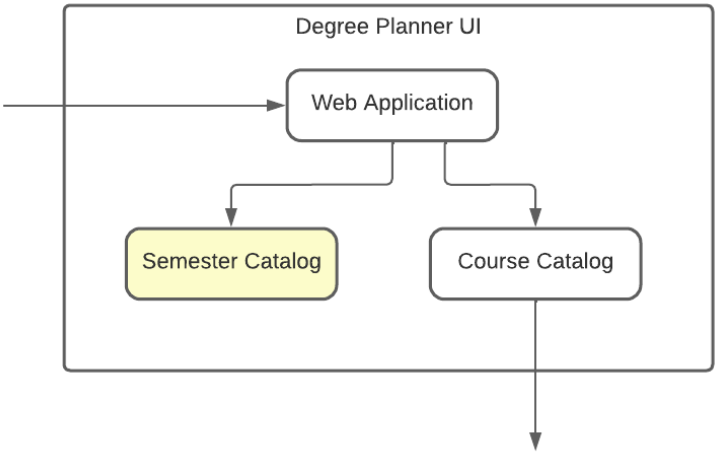


Figure 4: SemesterCatalog Subsystem

3.6.1 Subsystem Hardware

See section 3.1.

3.6.2 Subsystem Operating System

See section 3.2.

3.6.3 Subsystem Software Dependencies

See section 3.3.

3.6.4 Subsystem Programming Languages

This subsystem uses React with TypeScipt, and CSS.

3.6.5 Subsystem Data Structures

No particular data structure is required.

3.6.6 Subsystem Data Processing

This subsystem will read data and process the data into a Course object and send it to the web application system.

4 Firebase Layer

The Degree Planner Firebase Layer consists of two subsystems, Authentication and Firestore Database. It is reposible for authenticating the user and retrieving their information. The database function will return null unless the user has been authenticated with Firebase.

4.1 Layer Hardware

The Firebase layer is dependent on Firebase servers hosted by Google.

4.2 Layer Operating System

Google's Firebase does not publish what operating system their servers run on.

4.3 Layer Software Dependencies

The Firebase layer uses JavaScript/TypeScript to make requests to Firebase. It is dependent on the firebase-core package. This package is required for both subsystem's software dependencies.

4.4 Authentication Subsystem

Describe at a high level the purpose and basic design of this subsystem. Is it a piece of hardware, a class, a web service, or something else? Note that each of the subsystem items below are meant to be specific to that subsystem and not a repeat of anything discussed above for the overall layer.

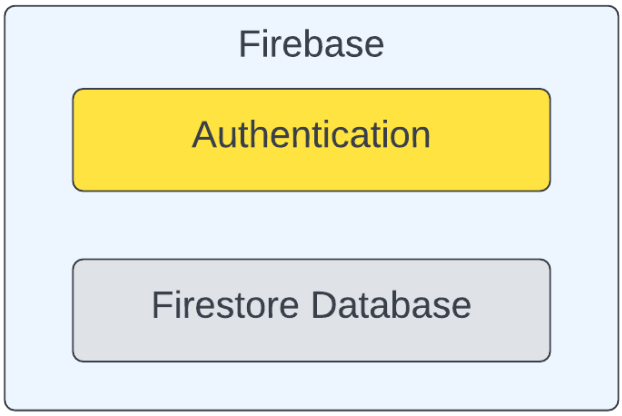


Figure 5: Diagram of the Firebase layer with the Authentication subsystem selected

4.4.1 Subsystem Software Dependencies

The Authentication Subsystem is dependent on the firebase-auth package to authenticate the user before they can use the website.

4.4.2 Subsystem Programming Languages

The Authentication Subsystem is programmed using TypeScript.

4.4.3 Subsystem Data Structures

The Authentication Subsystem has all authentication functions in one file, and each function is exported. This allows for the server to take advantage of tree shaking, which means only code needed is imported and not the entire file.

4.5 Firestore Database Subsystem

The Firestore Database Subsystem is responsible for retrieving user information, their chosen courses, and a list of all courses from the Firebase Firestore Database.

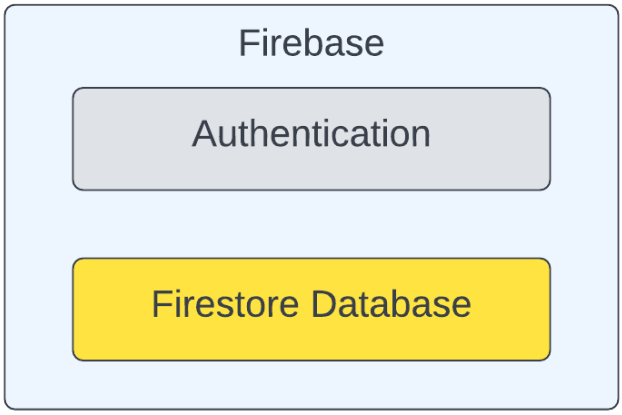


Figure 6: Diagram of the Firebase layer with the Firestore Database subsystem selected

4.5.1 Subsystem Software Dependencies

The Firestore Database Subsystem is dependent on the firebase-firestore package to make database queries to the Firestore Database.

4.5.2 Subsystem Data Structures

The Firebase Database Subsystem has all authentication functions in one file, and each function is exported. This allows for the server to take advantage of tree shaking, which means only code needed is imported and not the entire file.

4.5.3 Subsystem Data Processing

The Firebase Database Subsystem processes the data received from Firestore and converts it to Class Objects to take advantage of static analysis.

The subsystem converts the JSON string received from the all courses list in Firestore and converts it to a list of Courses objects.

The subsystem converts the user information stored in Firestore and converts it to the object User-Data. When processing the user information, it also converts the user's chosen classes to the Chosen-Course object. The ChosenCourse object is a minified version of Course with an id reference to the Course object with the full details.

5 Dashboard Layer

This is the dashboard subsystem where it will consist of a link to the Degree Planner, have the ability to view the flowchart of the chosen major, and will be able to sign in and sign out of the user account.

5.1 Layer Hardware

This is a pure software based project, hence no Hardware.

5.2 Layer Operating System

This is a web-based layer therefore any OS can be used to view it.

5.3 Layer Software Dependencies

This will be done using the REACT Typescript framework and the design will be done using CSS. Firebase authentication will be needed for user accounts.

5.4 Homepage/Login

Initially this page will look as the homepage, where the user will have the option to create an account or sign in. Once signed in, the page will turn into the page containing a degree flowchart along with a sidebar version of a dashboard which will have the option of signing out for users and include some user information.

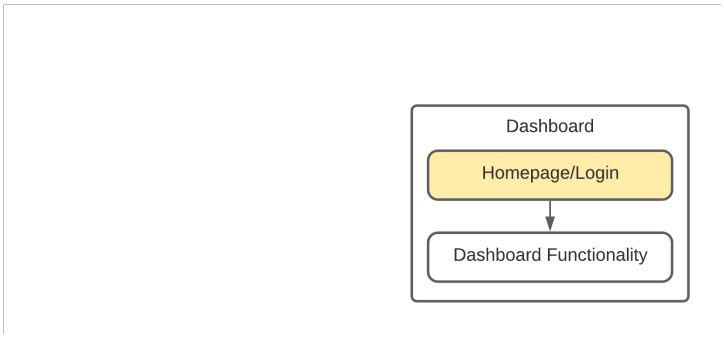


Figure 7: Homepage/Login Functionality

5.4.1 Homepage/Login Hardware

No Hardware is used.

5.4.2 Homepage/Login Operating System

No specific requirement for an Operating System.

5.4.3 Homepage/Login Software Dependencies

This is dependent on the Firebase authentication so user can sign up as well as Logging in to their account.

5.4.4 Homepage/Login Programming Languages

REACT Typescript and CSS are used.

5.4.5 Homepage/Login Data Structures

No specific data structures are used here since the login is based on user input and Firebase takes care of the rest.

5.4.6 Homepage/Login Data Processing

Data is not being processed in this section..

5.5 Dashboard

Initially this page will look as the homepage, where the user will have the option to create an account or sign in. Once signed in, the page will turn into the page containing a degree flowchart along with a sidebar version of a dashboard which will have the option of signing out for users and include some user information.

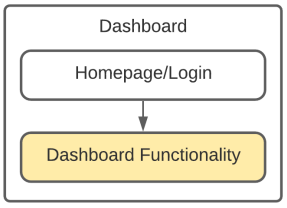


Figure 8: Dashboard Functionality

5.5.1 Dashboard Hardware

No Hardware is used.

5.5.2 Dashboard Operating System

No specific requirement for an Operating System.

5.5.3 Dashboard Software Dependencies

This dashboard will have a view flowchart feature where that feature is dependent on a react library called "react-zoom-pan-pinch".

5.5.4 Dashboard Programming Languages

REACT Typescript and CSS are used.

5.5.5 Dashboard Data Structures

No specific data structures are used here.

5.5.6 Dashboard Data Processing

Data is not being processed in this section.

6 APPENDIX A

Include any additional documents (CAD design, circuit schematics, etc) as an appendix as necessary.

REFERENCES