COMP4960 – Software Engineering
Architectural Design

Campus Lost and Found Portal

Team Name, ID: Team 2
Date: 11/03/2025

Outline

- Introduction
- Requirements
- System Architecture
  - Architecture Pattern(s)
  - Overall Architecture
  - Components Mapping
- Technology Stack Selection

Introduction

- Our application is a web and iOS application that allows user to submit reports for items they either lost or found around campus.
- Users can attach images to their reports to help users with matching lost and found items.
- Users can report where an item was lost or found, including the Wentworth hall name and a map pin.
- Administrators can moderate the submitted reports to ensure there is no inappropriate content uploaded to the publicly visible app.

Requirements

Functional

- User registration via Wentworth Microsoft account.
- Search bar actions for user to search through list of found items.
- View, edit or delete options for user reports.
- Report a lost or found item(category, description, location, and contact information).

Non-Functional

- Access restricted to only @wit.edu accounts.
- All reports shall be saved in Firebase database.
- All reports shall require item description, category, and location.
- Lost and found reports shall be approved by an administrator, before they are displayed in the app.

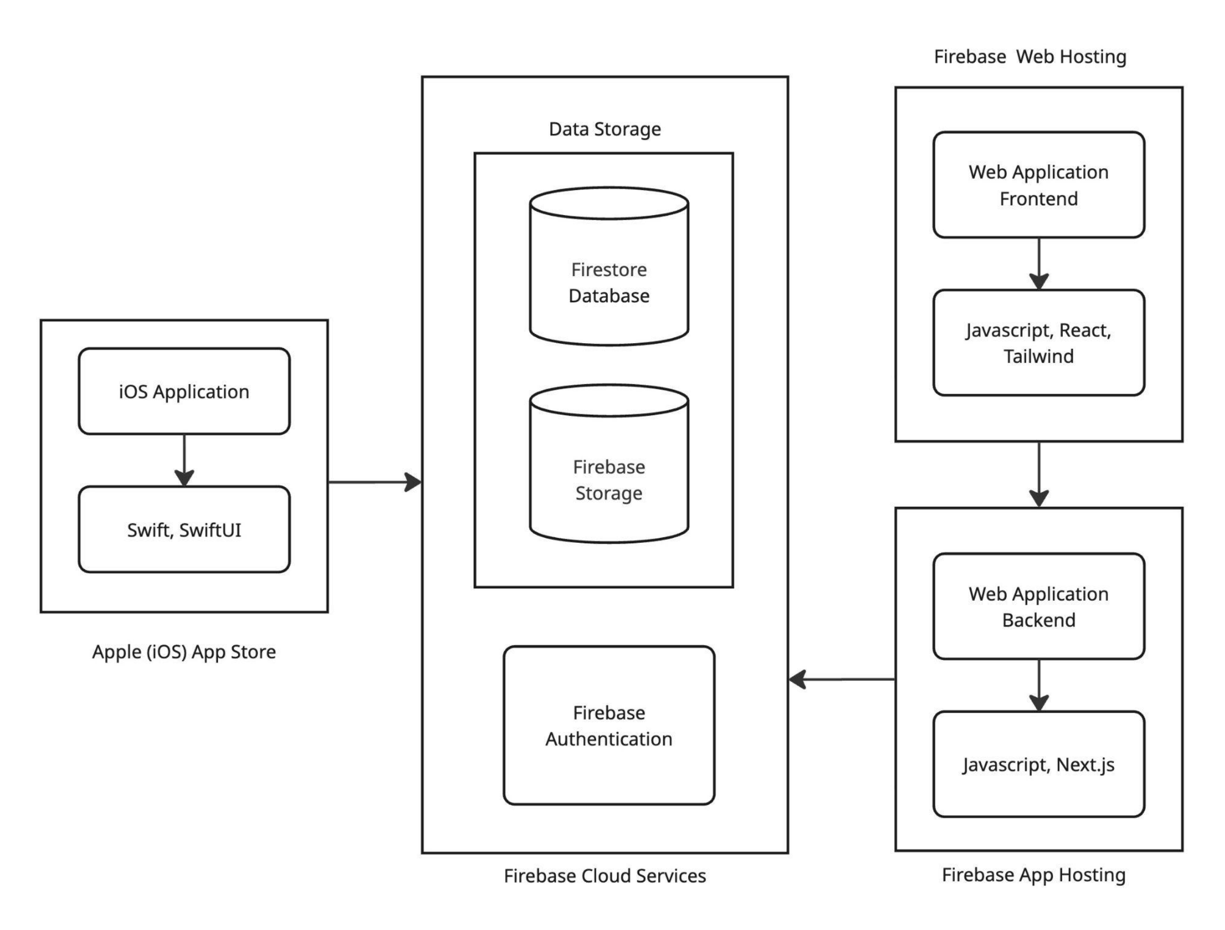
Architectural Pattern

Pattern(s): Model-View-Controller (MVC)
Because:
1) Ideal pattern to use for web/mobile applications

2) Items will be stored as data in the model. The Controller takes the requests of the user that interacts with the view.

3) The simple back and forth requests from user to controller is what makes this pattern ideal.

Overall Architecture



Explanation

1. **iOS Application:** Users can submit lost and found item reports, and view approved reports from other users.
2. **Web Application Frontend:** *Basic* users can submit lost and found item reports, and view approved reports from other users. *Administrators* can approve and deny user-submitted item reports.
3. **Web Application Backend:** acts as a middleman for API calls send by the frontend.
4. **Firebase Authentication:** Allows users to sign in using their wit.edu Microsoft account.
5. **Firestore Database:** Stores user reports and other general text-based app data.
6. **Firebase Storage:** Stores user-submitted images.

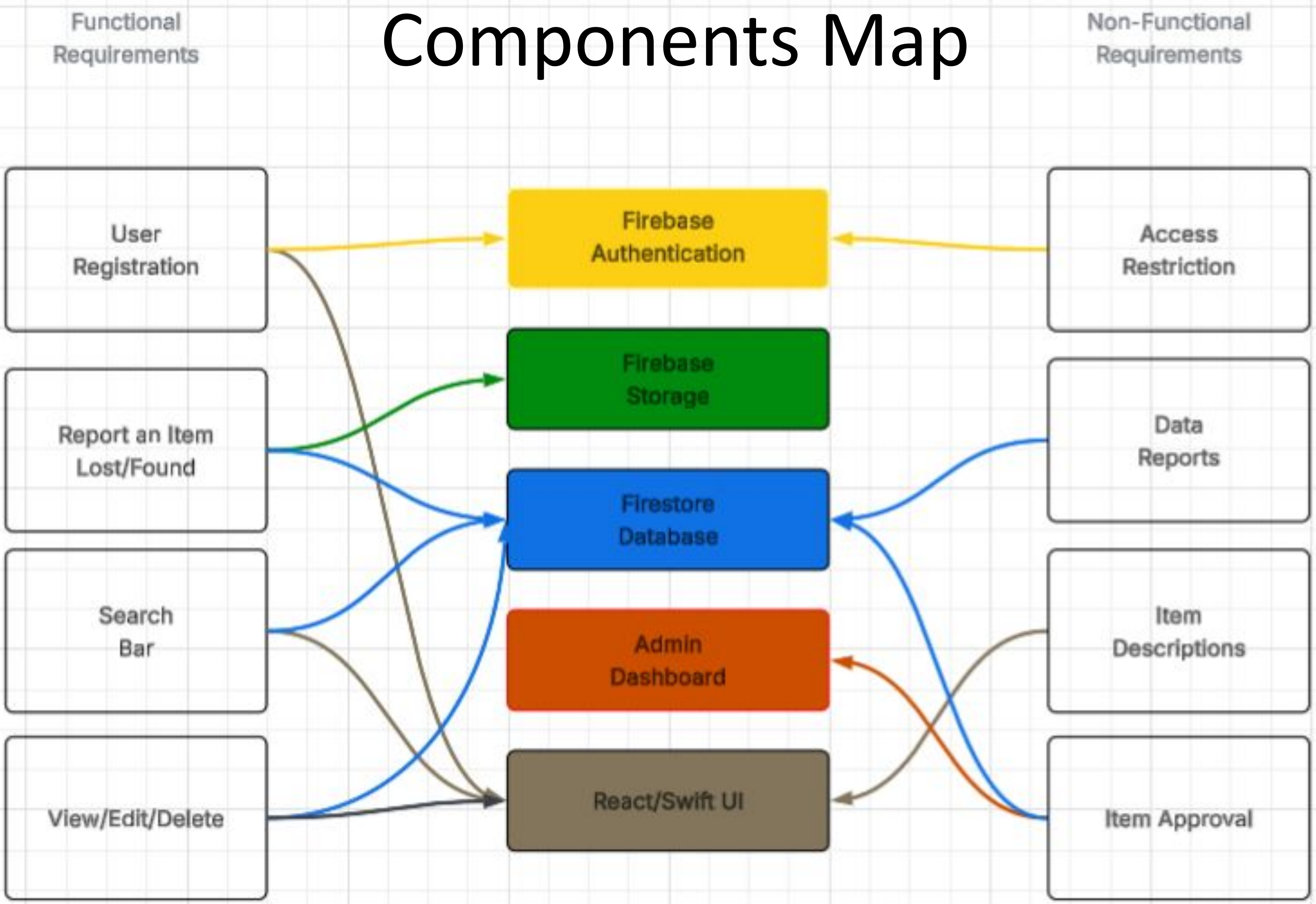
Components Mapping for Functional Requirements

Functional requirements	**Mapped components**
User registration via Wentworth Microsoft account.	Firebase Authentication
Search bar actions for user to search through list of found items.	React/SwiftUI, Firestore Database
View, edit or delete options for user reports.	React/SwiftUI, Firestore Database
Report a lost or found item(category, description, location, contact information, and optional image).	React/SwiftUI, Firestore Database, Firebase Storage

Components Mapping for Non-Functional Requirements

Non-functional requirements	**Mapped components**
Access restricted to only @wit.edu accounts.	Firebase Authentication
All reports shall be saved in the database.	Firestore Database
All reports shall require item description, category, and location.	React/SwiftUI
Lost and found reports shall be approved by an administrator, before they are displayed in the app.	Firestore Database, Admin Dashboard

Components Map



Technology Stack Selection

Technology	Purpose
Firestore Database	Stores user reports and other general text-based app data.
Firebase Storage	Stores user-submitted images attached to lost/found reports.
Firebase Auth	Manages user sign-in, restricted to @wit.edu Microsoft accounts.
Firebase Hosting	Hosts the frontend and backend of the web application.
Javascript	Programming language that easily integrates required libraries for both frontend and backend.
Tailwind	CSS framework for rapidly styling the web application's frontend user interface.
React	JavaScript library for building the web app’s interactive UI components and dynamic pages.
Next.js	Framework used for server-side logic, routing, and acting as the API middleman
Visual Studio Code	Development environment used to build the web application’s frontend and backend.
Xcode	Development environment for developing, debugging, and compiling the iOS application.
Swift/SwiftUI	Programming language and native UI kit for building the iOS app.
Github and Jira	Development and collaboration tools used for tracking project progress.
Figma	Design tools used for UI mockups and visual diagrams like flowcharts and architecture.

Q&A