



Diverse Makers

Software Design Document

Version 1.0

09-20-2024

Project Sponsor: Dr. Jared Duval

Faculty Member: Isaac Shaffer

Project Mentor: Vahid Nikoonejad Fard

Team members:

Daniel Minichetti (Team Lead)

Kane Davidson

Eduardo De La Rosa

Elleana Negrelli

Aaron Ramirez

Overview

The purpose of this document is to discuss the overall architecture design of our project framework. It will encompass all the important components and design choices required to build an app that is as accessible to a majority of people with varying disabilities.

Table of Contents

1.0 INTRODUCTION	2
2.0 IMPLEMENTATION OVERVIEW	4
2.1 Solution Vision	4
2.2 Technologies	5
3.0 ARCHITECTURAL OVERVIEW	7
3.1 User Interface (UI) Component	7
3.2 Business Logic Component	8
3.3 Data Access Component	8
3.4 External Service Handler	8
4.0 MODULE AND INTERFACE DESCRIPTIONS	10
4.1 User Interface (UI) Component	10
4.2 Business Logic Component	11
4.3 Data Access Component	13
5.0 IMPLEMENTATION PLAN	15
6.0 CONCLUSION	16

1.0 INTRODUCTION

Diversity is an important subject in science and technology. Leveraging different backgrounds, experiences, and points of view enriches STEM and brings unique insights to solving problems and making breakthroughs. Over 40 million Americans have a disability; however, research shows that disabled people are severely underrepresented in STEM fields, so much so that only 3% of people in the STEM workforce have a disability. People with disabilities tend to learn best in hands-on and individual learning environments, which can be difficult to accommodate in today's education system. Makerspaces can provide the resources to create this environment for STEM learning.

It is apparent from the workforce and college statistics that there are clear underrepresentation and support systems to aid those with disabilities in a career in STEM. Dr. Jared Duval aims to increase awareness and opportunities for those with disabilities with a social app designed to connect makerspaces to people of all disability backgrounds.

Dr. Jared Duval is an assistant professor at the School of Informatics, Computing, and Cyber Systems at Northern Arizona University. He is also the director of the Playful Health Technology Lab, an interdisciplinary research team whose mission is to design, implement, and study human-computer interaction technologies that enhance the experience of improving and maintaining one's health. Dr. Duval utilizes research through design to develop therapy games and playful applications that make health more motivating and sustainable.

Additionally, Dr. Duval's work specializes in serious games for health that emphasize human-computer interaction with assistive technology. Working with Dr. Duval and prioritizing accessibility, we can empower those with disabilities to explore their passions and aspirations in STEM by creating a collaborative and supportive environment built around makerspaces.

The design of our application will cater to a large group of users who wish to progress their STEM education. In providing STEM learning content and resources to individuals with a variety of disabilities, our top priority is to provide these materials in the most accessible way possible. Taking a more in-depth look into our project, the following table covers several key user-level requirements that will be integrated into our application. Further, some vital

functional/performance requirements will be addressed with the environmental constraints observed throughout our development process to implement solutions to the many challenges facing the makerspace community.

User Level Requirements	Functional Requirements	Performance Requirements	Environmental Constraints
<ol style="list-style-type: none"> 1. Enable user profile creation and management 2. Provide accessible STEM material and resources 3. Connect users to local makerspaces near them 4. Facilitate communication between users and local makerspaces 5. Allow makerspaces to upload and share STEM content and material to the platform 	<ol style="list-style-type: none"> 1. Intuitive account creation and management 2. Secure user authentication and login system 3. Resource hosting for STEM learning content 4. Accessible user interface for multiple disabilities 5. Location-based makerspace discovery 6. Messaging system for makerspace communication 7. Search and filter functionality for STEM content 	<ol style="list-style-type: none"> 1. Cross-platform compatibility for users on either IOS or Android 2. Optimized data retrieval for quick access to content 3. Responsive app layout for different devices and orientations 	<ol style="list-style-type: none"> 1. Backend development with Google's Firebase technology 2. Compliance with modern accessibility standards 3. Adherence to user privacy regulations and app usage

In this document, we will delve into the deeper accessibility and development concerns involving the project, and then propose a detailed solution to address these issues. In each subsequent section, we will explore the architectural design of our app and describe each major module within our system. Additionally, we will outline our implementation plan and explain how our solution will be realized for use within the makerspace community. With these goals in mind, we hope to transform the makerspace landscape and bring a crucial tool that everyone can use to better their STEM education learning outcomes and pursue their passions.

2.0 IMPLEMENTATION OVERVIEW

2.1 Solution Vision

To develop an application that provides as much accessibility to the user as possible, our target is to create a framework that heavily follows web content accessibility guidelines without compromising on sharing local makerspace resources with those who may not have physical access to a local makerspace or succeed in learning better in a remote environment. Some features we want to implement in our framework include a system for makerspaces to upload resources onto the application and a user-end system to view content and manage account settings. Our framework's success will be measured by creating an ecosystem to access STEM content that is both fluid and efficient for both our users and makerspaces providing learning resources. The main objective that our app, Diverse Makers, is trying to achieve is to create an accessible framework tailored to makerspaces and their clients that will expand STEM education and expedite processes.

To address the previously stated problems, we will create a crowd-sourced mobile application that will act as a centralized hub for sharing STEM resources, projects, and training for those with disabilities. This application will connect people with makerspaces and allow them to easily access projects, resources, and information that makerspaces provide. Our application will bridge the gap and provide a number of services to users, including:

- Hosting STEM-related resources, accommodations, community contributions, and processes.
- Utilizing makerspace resources to create community frameworks for those with disabilities.
- Creating a network to connect makerspaces and support discourse around these resources.

Our application will work closely with makerspaces to provide relevant resources. In addition to this, makerspaces will be able to upload their own resources to be hosted on the application; these resources could be a video, picture, or text. While this introduces more work for makerspaces, the trade-off is worth it, especially if it causes STEM resources to become more accessible to those who need them.

2.2 Technologies

Our Diverse Makers application will utilize a combination of cutting-edge technologies to ensure that we produce a robust, scalable, and highly accessible platform for all users, regardless of their abilities. The key technologies we will employ are:

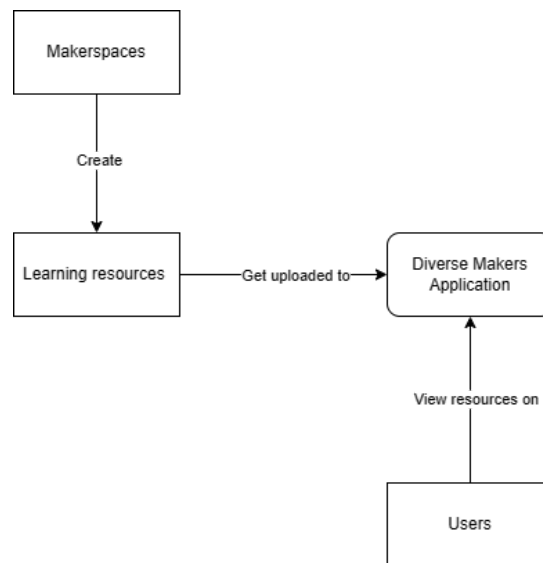
1. **Backend:** Firebase, Google's mobile and web application development platform, will serve as our backend-as-a-service (BaaS) platform solution. We've chosen it for several reasons:
 - Real-time database: Allows for instant data synchronization across all clients.
 - Authentication: Offers secure and easy-to-implement user authentication.
 - Cloud Functions: Enables serverless computing power for backend operations.
 - Hosting: Provides fast and secure hosting for our web content.
 - Affordable: For our use case, it will be free.
2. **Frontend:** React Native, a popular open-source mobile application framework, will be used to develop our cross-platform mobile application. We've chosen it for the following reasons:
 - Cross-Platform Development: Allows us to build both iOS and Android apps from a single codebase.
 - UI/UX Design: Utilizing React Native Paper, UI components within the app will adhere to Material Design guidelines, ensuring accessibility and usability.
 - Performance: Offers near-native performance through its optimized components. Though it has limited access to some device hardware, we won't need it for this application.
 - Large Community and Ecosystem: Provides access to many libraries and tools that should simplify the development process. Additionally, there's a large quantity of learning resources that will help us.
3. **Accessibility Frameworks and Guidelines:** To ensure our application is accessible to the widest possible audience, we will adhere to the following standards:

- Web Content Accessibility Guidelines (WCAG) 2.1 focuses on the four main principles:
 - Perceivable: Information must be presented to users in ways they can perceive.
 - Operable: User interface elements must be operable by all users with different types of disabilities.
 - Understandable: Information and operation must be understandable.
 - Robust: Content must be robust enough to be interpreted by a wide variety of user agents.
- Nielsen's ten Usability Heuristics will be incorporated to enhance overall usability. These include:
 - Visibility of system status: Users should always be informed of what is going on through appropriate feedback.
 - Match Between the System and the Real World: Design should speak the user's language.
 - User Control and Freedom: Users often perform actions by mistake. They need a clear, easy way to back out of something.
 - Consistency and Standards: Users should not have to worry whether different words, situations, or actions mean the same thing.
 - Recognition rather than recall: Minimize the user's memory load by making elements, actions, and options always visible.
- We will also utilize the React Native Accessibility library and APIs to implement our accessibility features.
 - Accessibility API: Ensure our UI components are accessible and accommodate various disabilities.
 - Assistive Technologies: Enables our application to include screen readers functionality natively with VoiceOver (IOS) and TalkBack (Android).

By combining these technologies and adhering to accessibility guidelines, we aim to create an application that is not only powerful and efficient but also inclusive and usable by individuals with a wide range of abilities. This approach aligns with our primary goal to make

STEM resources and Makerspaces accessible to all, regardless of ability. In the process, we hope this makes STEM a more welcoming place for everyone.

3.0 ARCHITECTURAL OVERVIEW



(a) Key Responsibilities of Each Component

3.1 User Interface (UI) Component

- Key Responsibility: Provide an accessible, intuitive interface for all users.
- Features:
 - Responsive design for iOS and Android devices
 - Customizable accessibility (E.g. High Contrast modes, adjustable font sizes.)
 - User Profile Management
 - Content browsing and search functionality

3.2 Business Logic Component

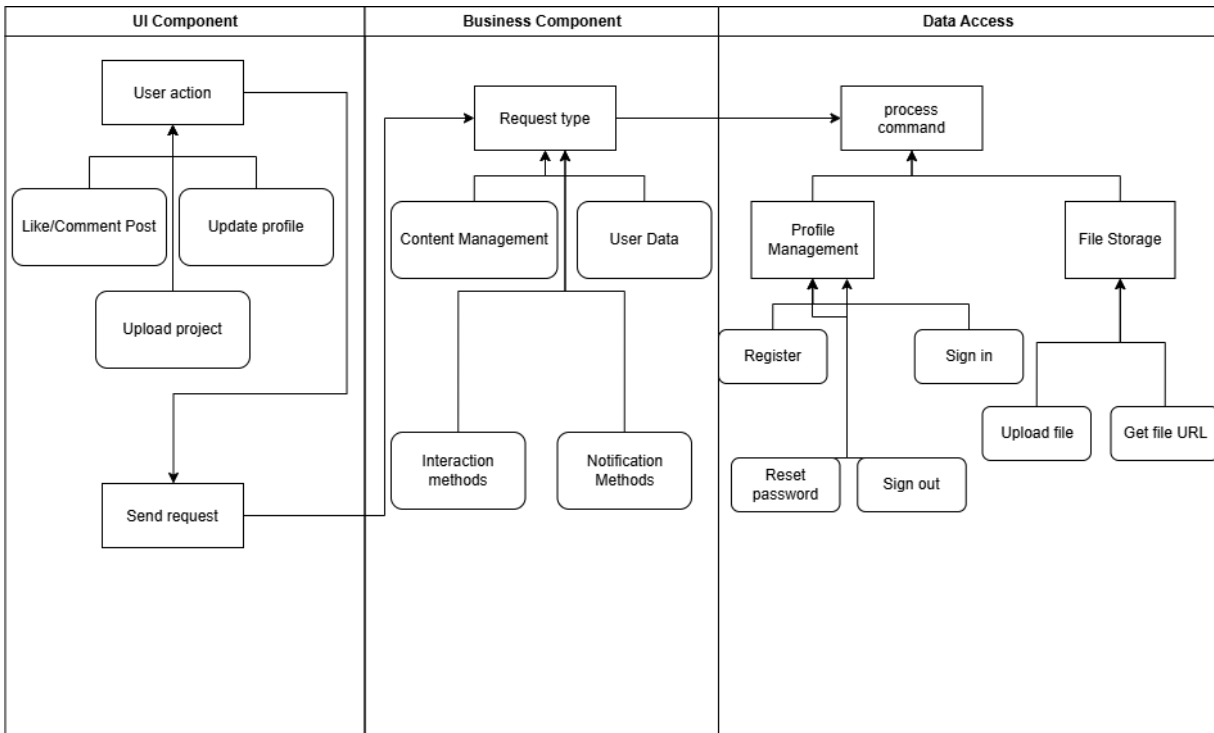
- Key Responsibility: Handle core application functionality and data processing
- Features:
 - User Authentication and Authorization
 - Makerspace Matching based on user location, preferences, etc
 - Resource management (upload, categorization, tagging)

3.3 Data Access Component

- Key Responsibility: Manage data storage and retrieval operations
- Features:
 - Integration with Firebase real-time database and Cloud Firestore
 - Caching mechanisms
 - Data synchronization

3.4 External Service Handler

- Key Responsibility: Integrate and communicate with third-party services.
- Features:
 - Push notification system for updates and messages
 - Geolocation services for makerspace discovery (if time permits)



(b) Communication & Control Flow between Components

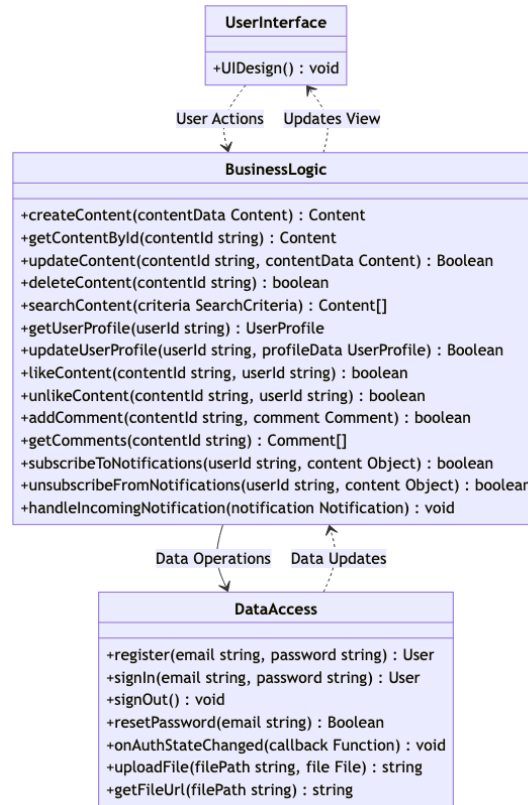
1. Client-Server Communication:

- React Native app communicates with the Firebase backend using Firebase SDK
- RESTful API calls for data retrieval and updates
- Realtime data synchronization using Firebase real-time database

(c) Influence from Architectural Style

1. Layered Architecture pattern

- Separates the application into Model (Data Access), View (UI), and Controller (Business Logic) components, but in a top down way where a layer can only interact with the layer beneath it.
- Enhances maintainability and allows us to update components more easily.

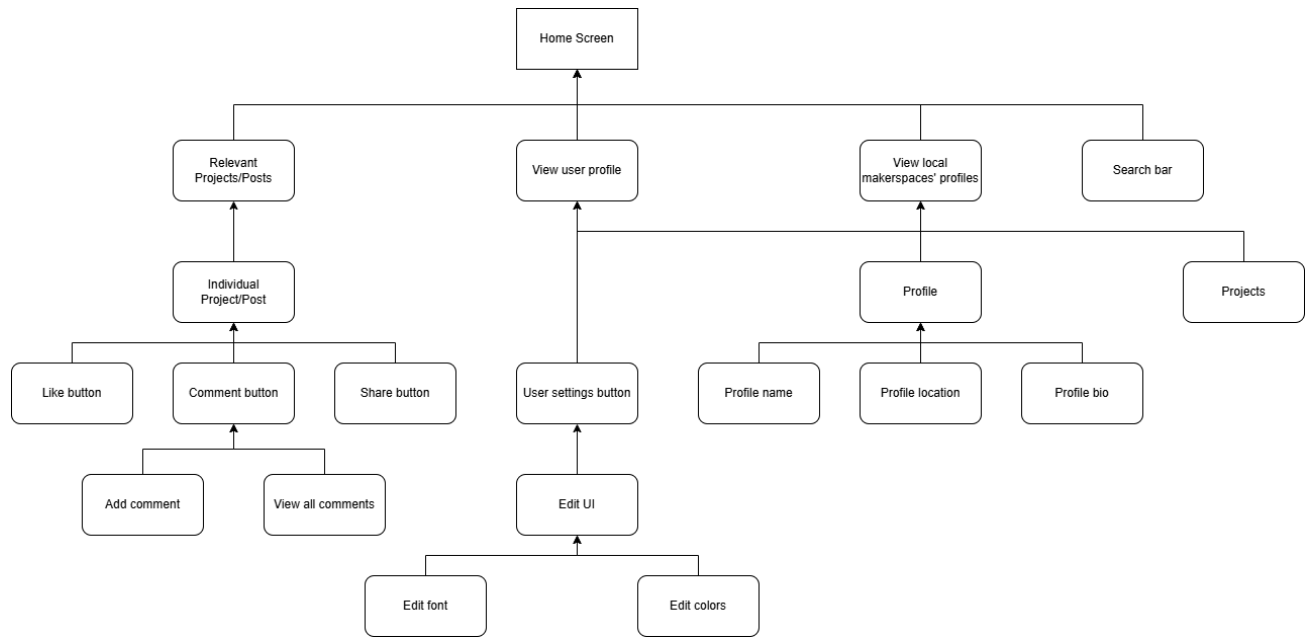


4.0 MODULE AND INTERFACE DESCRIPTIONS

4.1 User Interface (UI) Component

The User Interface (UI) should provide an intuitive experience that is accessible to all users of our application. We aim to achieve this by providing users with high customization of their interface such as changing font sizes, providing a high contrast mode, and other control over their experience.

4.1.1 Public Methods



As seen in the diagram above, the UI will consist of a home screen, which is where the user can access other components of the application. Firstly, they will be able to view their relevant projects. Each of these projects will have their own like, share, and comment button. The comment button will allow them to add and view other comments for the respective post. Back to the home screen, the user will be able to view their own profile where they can change things such as their name, location, bio, projects they have participated in, and personal settings in the application. These personal settings include the font, colors, and other key attributes of the application UI's appearance. In addition to this, they will also be able to view makerspaces' profiles, which will also have their name, location, bio, and the projects that the makerspace has uploaded. These UI components will ensure our application offers an intuitive and highly customizable user experience.

4.2 Business Logic Component

The Business Logic Component manages the core functionalities of our application. This includes processing user input, facilitating secure authentication, managing makerspace matching, and handling resource uploads. The business logic and data access components work

together to retrieve and store data, bridging the user interface to the backend and enabling seamless communication between the two for optimal user experience.

4.2.1 Public Methods

- Content Management:
 - `createContent(contentData: Content): Promise<Content>`
 - `getContentById(contentId: string): Promise<content>`
 - `updateContent(contentId: string, contentData: Content): Promise<Boolean>`
 - `deleteContent(contentId: string): promise<boolean>`
 - `searchContent(criteria: SearchCriteria): Promise<Content[]>`
- User Data Methods:
 - `getUserProfile(userID: string): Promise<UserProfile>`
 - `updateUserProfile(userId: string, profileData: UserProfile): promise<Boolean>`
- Interaction Methods:
 - `likeContent(contentId: string, userId: string): Promise<boolean>`
 - `unlikeContent(contentId: string, userId: string): Promise<boolean>`
 - `addComment(contentId: string, comment: Comment): Promise<boolean>`
 - `getComments(contentId: string): Promise<Comment[]>`
- Notification Methods:
 - `subscribetoNotifications(userId: string, content: {TBD}): Promise<boolean>`
 - `unsubscribeFromNotifications(userId: string, content{TBD}): Promise<boolean>`
 - `handleIncomingNotification(notification: Notification): void`

BusinessLogic
+createContent(contentData Content) : Content +getContentById(contentId string) : Content +updateContent(contentId string, contentData Content) : Boolean +deleteContent(contentId string) : boolean +searchContent(criteria SearchCriteria) : Content[] +getUserProfile(userId string) : UserProfile +updateUserProfile(userId string, profileData UserProfile) : Boolean +likeContent(contentId string, userId string) : boolean +unlikeContent(contentId string, userId string) : boolean +addComment(contentId string, comment Comment) : boolean +getComments(contentId string) : Comment[] +subscribeToNotifications(userId string, content Object) : boolean +unsubscribeFromNotifications(userId string, content Object) : boolean +handleIncomingNotification(notification Notification) : void

4.3 Data Access Component

The Data Access component is responsible for all interactions with the Firebase backend. It serves as an abstraction layer between the Business Logic Component and the database. It has the following responsibilities:

- **Authentication Management:** handle user sign-up, login, logout, password reset, authentication state changes, and other related tasks using Firebase authentication.
- **Database Operations:** Perform CRUD (Create, Read, Update, Delete) operations on user-generated content, profiles, and other data using Firebase Firestore.
- **File Storage:** Manage uploading and retrieving media files (e.g., images, documents) with Firebase Storage.
- **Real-time Updates:** Listen for and propagate real-time changes to keep the app content up to date.
- **Security Enforcement:** Apply Firebase security rules to restrict unauthorized access and protect user data.

4.3.1 Public Methods

- Authentication Methods: May not be needed if we opt to only use OAuth
 - register(email: string, password: string...): Promise<user>
 - signIn(email: string, password: string): Promise<User>
 - signOut(): Promise<void>
 - resetPassword(email: string): Promise<Boolean>
 - onAuthStateChanged(callback: Function): void
- File Storage Methods:
 - uploadFile(filePath: string, file: File): Promise<string>
 - getFileUrl(filePath: string): Promise<string>

DataAccess
+register(email string, password string) : User +signIn(email string, password string) : User +signOut() : void +resetPassword(email string) : Boolean +onAuthStateChanged(callback Function) : void +uploadFile(filePath string, file File) : string +getFileUrl(filePath string) : string



6.0 CONCLUSION

In conclusion, our mobile application aims to increase diversity in STEM by not only connecting individuals with disabilities to makerspaces but providing crucial STEM content that will increase their learning outcomes. In collaboration with Dr. Jared Duval, we will enable those who have previously been underrepresented and lacked a support system, help discover and obtain a career in STEM. Bringing awareness to the challenges facing those with disabilities, our team will transform the makerspace community by creating a centralized hub for makerspaces to share content, providing users with the tools to bridge a gap in their STEM education.

In creating a comprehensive plan to bring this vision to life, our application and system architecture will prioritize accessibility to provide a seamless and intuitive experience for users with various disabilities. From the secure authentication process of sign up to the accessible STEM content on the platform, every component will be designed to meet the needs of our users. With this approach in mind, we will utilize technologies like React Native and Google Firebase to create an app that supplements the learning process and gives users the opportunity to interact with content that broadens their learning while connecting them with their local makerspace.

As we move forward into the implementation stage, our team will follow the milestones outlined in the Gantt chart to make sure that each feature is complete and performs its intended functionality within the overall system. Each member of the team will work in conjunction with one another to complete the requirements needed in the frontend and backend modules. We anticipate the start of our development process to begin with the front end design as accessibility within the app is vital. We will create the overall design and layout for the app, ensuring it satisfies our requirements for accessibility and usability for a variety of disabilities. In conjunction with frontend development, the backend system will be built to enable actions taken by the user to be reflected in our database.

To combat any potential missteps in our development and implementation plan, our team is committed to meeting with our Dr. Duval to communicate any issues and uncertainties we may encounter as we put each major component together. Together with weekly meetings to reflect on

our progress, our team is prepared and ready to make a significant positive contribution towards STEM accessibility, paving the way for a more diverse STEM community.