Placenet

for loved places & people

Devlin **Hicks**, Ethan **Hoffman**, William **Robinson**, Shaun **Whitlow**, Angeles **Marin Batana**

Client & Team Info

Client:

- Kazem Namazi
  - Architect based in Indianapolis
    - Entrepreneur & Creator of Placenet

Team:

- Devlin Hicks
  - Handle Canvas Submissions
    - C.S Major
- Ethan Hoffman
  - Virtual Meetings with Mentor
    - C.S Major
- William Robinson
  - Note Taker During Meetings
    - C.S Major, Game Design and Development Conc.
- Shaun Whitlow
  - Canvas Submissions
    - C.S Major
- Angeles Marin Batana
  - Client Interaction
    - C.S Major, Data Analytics & Machine Learning Conc.

Placenet

- Mobile application design to simplify property management & encourage homeowners and tenants to take proactive care of their homes.

    - **GOAL**:
        - Make managing & improving properties easier

William- Business Requirement

Business Requirements

BR1

The app will improve property care for homeowners and tenants by providing user friendly property management tools. Users will be able to add/remove owned properties to/from a profile, update and edit projects that have been done to a property, upload documents to show proof of work, and edit user profiles.

Why this is a requirement:

- It covers the full functionality of the project and what it's capabilities are.
- It explains the functionality in a manner that is neither too general nor too specific.
- Itemizes each function in an easy to read manner.

Ethan- Requirements

Requirements

FR1. User Authentication

- Sign up/ in
  - Users will be able to create an account by entering
    - Email
    - ***Password***

FR2.. Main Page/ Menu

- Add/ edit property
- Add/ edit project
- Add/edit documents

FR3. Property Management

- add/ edit property

users will be able to add/ edit/ update a property on their account with the following data:

- Property location:
  - Address
  - City
  - State
  - Zip

FR4. Project Management

- add/ edit project

User will be able to add/ edit a project on their account with the following data:

- Work done to property
  - Item name
  - Title of what was done
  - Date completed
  - Contractor information

FR5. Document Management

Users will be able to upload pictures of their documents

- Upload receipt/ invoice
- Attach document to property
- Edit basic data
  - Name of document
  - Date added

FR6.. User Profile settings

Users will be able to update/ edit their info

- Can change
  - Username
  - Email
  - Password

Devlin- 1st Iteration Requirements

Features For First Iteration:

1. User Authentication:
    a. Email
    b. Password
2. Main Page And Menu:
    a. Add or Edit Property
    b. Add or Edit Project
    c. Add or Edit Documents
3. Property Management:
    a. Address
    b. City
    c. State
    d. ZIP Code
4. Project Management:
    a. Item Name
    b. Title of project
    c. Completion Date
5. Document Management:
    a. invoice/receipt upload
    b. Property document attachment
    c. Basic Data Editing:
        i. Document Name
        ii. Date

Ethan- Use Cases

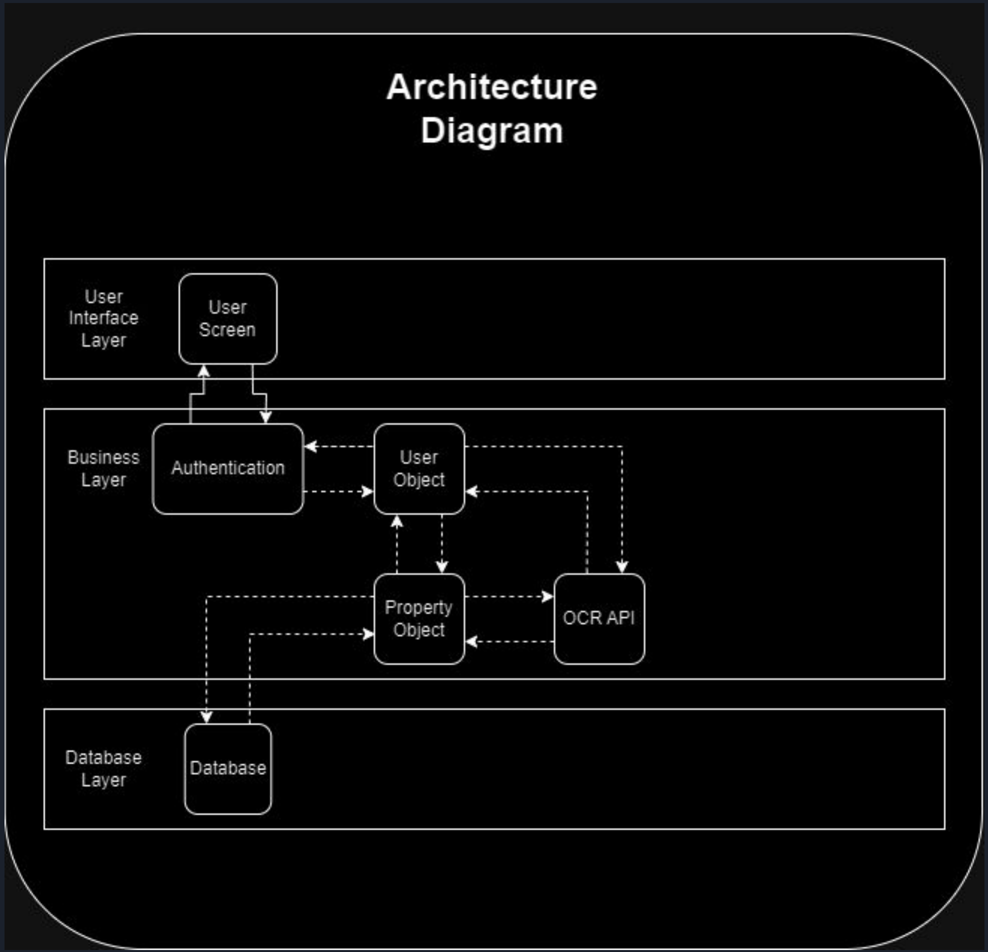
Use Case

Actor: Homeowner
Use case: As a homeowner, I can create an account on the app by entering my email and creating a password. This is necessary as user authentication and security is a key aspect of a mobile application.

- A homeowner will create an account by:
  - Entering an email
  - Creating a password

This points to the business requirement BR1 by ensuring a secure manner that users can access and use the app.

William- Architecture


Architecture Diagram
User Interface Layer
User Screen
Business Layer
Authentication
User Object
Property Object
OCR API
Database Layer
Database

Shaun- Domain Model

Placenet Domain Model Overview

Description: Illustrates key entities of our application which helps homeowners and tenants manage property maintenance through structured interactions between the users, properties projects and documents.

User: Manages a profile and connects to multiple properties.

Property: Represents a real estate asset.

Project: A task or job associated with one or more properties.

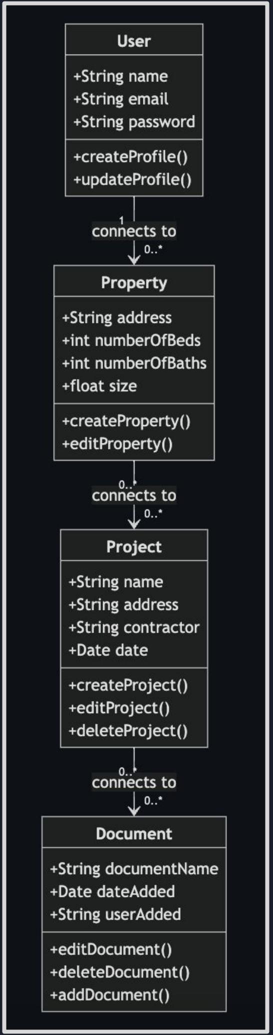
Document: Files uploaded to projects for better property management.

Key Relationships and Features

- User ↔ Property: Users can manage multiple properties (one-to-many relationship).

- Property ↔ Project: Properties can be linked to multiple projects, and each project can be linked to multiple properties (many-to-many).

- Project ↔ Document: Projects can involve multiple documents, which are maintained and edited over time (many-to-many).



User

+String name
+String email
+String password

+createProfile()
+updateProfile()

Property

+String address
+int numberOfBeds
+int numberOfBaths
+float size

+createProperty()
+editProperty()

Project

+String name
+String address
+String contractor
+Date date

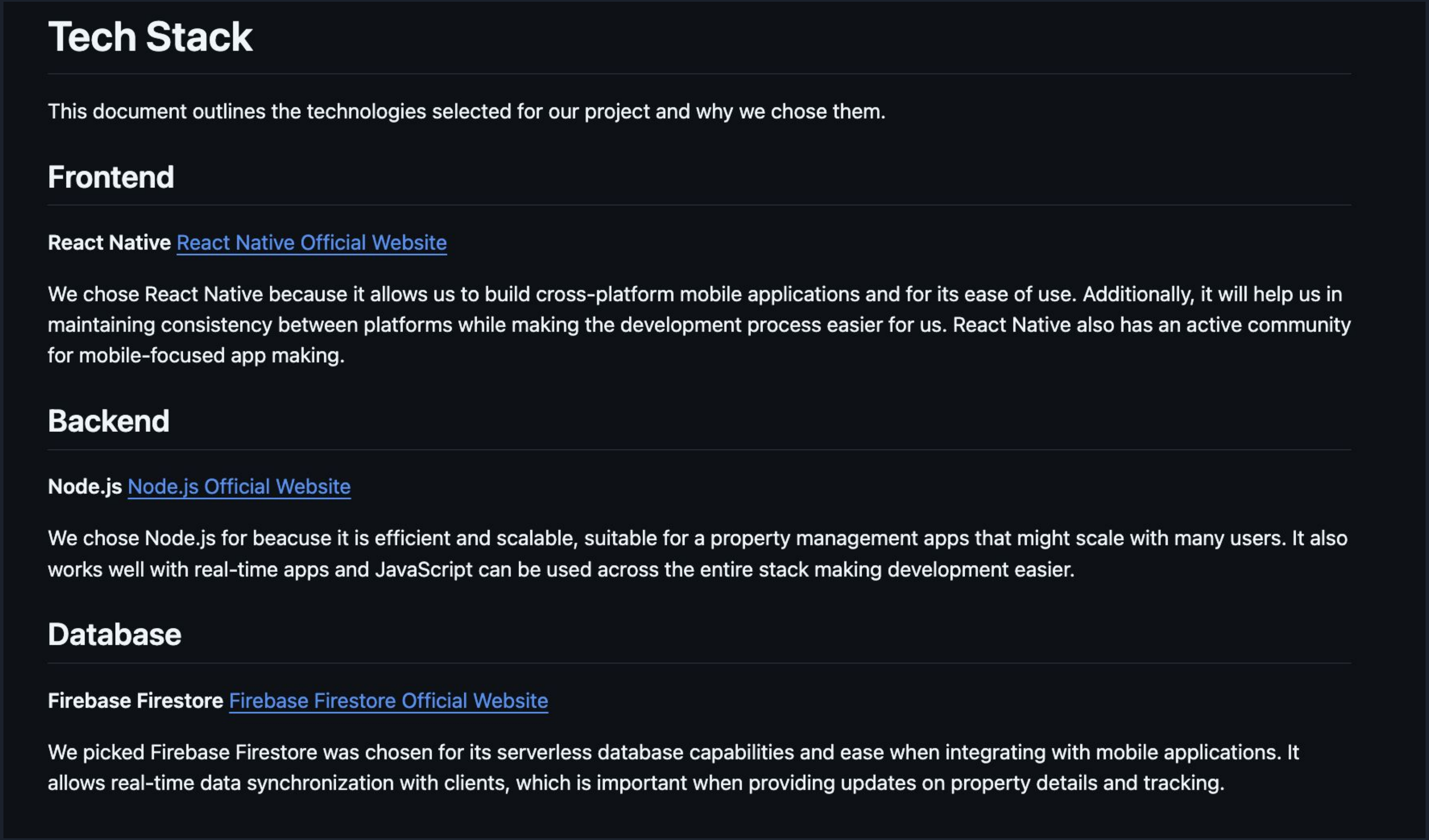
+createProject()
+editProject()
+deleteProject()

Document

+String documentName
+Date dateAdded
+String userAdded

+editDocument()
+deleteDocument()
+addDocument()

Angeles- Tech Stack



Tech Stack

This document outlines the technologies selected for our project and why we chose them.

Frontend

React Native React Native Official Website

We chose React Native because it allows us to build cross-platform mobile applications and for its ease of use. Additionally, it will help us in maintaining consistency between platforms while making the development process easier for us. React Native also has an active community for mobile-focused app making.

Backend

Node.js Node.js Official Website

We chose Node.js for beacuse it is efficient and scalable, suitable for a property management apps that might scale with many users. It also works well with real-time apps and JavaScript can be used across the entire stack making development easier.

Database

Firebase Firestore Firebase Firestore Official Website

We picked Firebase Firestore was chosen for its serverless database capabilities and ease when integrating with mobile applications. It allows real-time data synchronization with clients, which is important when providing updates on property details and tracking.

Tech Stack

Frontend:

- React Native
  - Cross- platform; ease of use

Backend

- Node.js
  - Efficiency

Database

- Firebase
  - all-in one

Angeles- Prototype

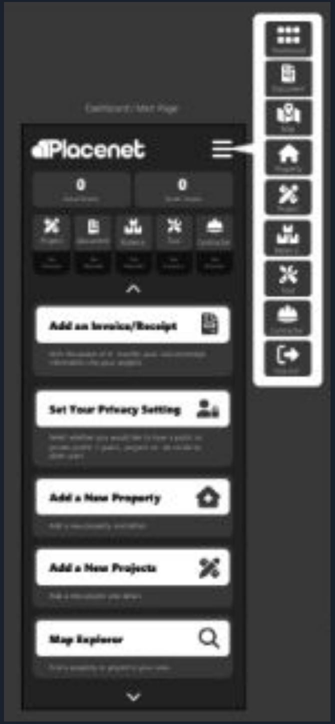
Prototype

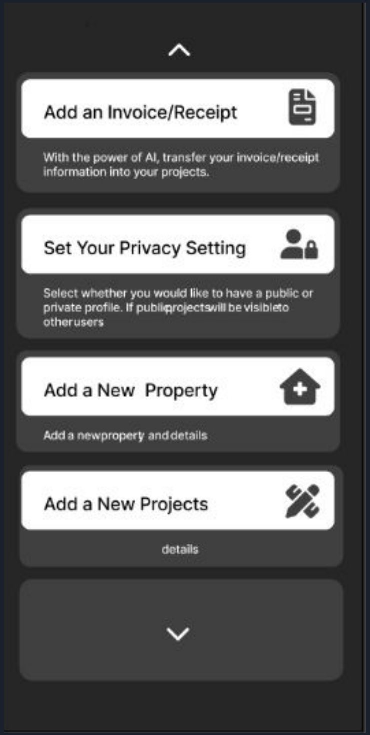
Client Prototype :

- [Client]

Our Prototype:

- [Ours]




Add an Invoice/Receipt

With the power of AI, transfer your invoice/receipt information into your projects.

Set Your Privacy Setting

Select whether you would like to have a public or private profile. If publicprojectswill be visibleto otherusers

Add a New Property

Add a newproperty anddetails

Add a New Projects

details