Knockk

Grace Radlund

CST-451 Capstone Project Requirements Document

Grand Canyon University

Instructor: Professor Mark Reha

Revision: 1.4

Date: 4/24/2025

**ABSTRACT**

Although social media is prominent, there is no social network to connect residents of an apartment building. This project aims to fill that void with Knockk. Knockk is a social network composed of two applications: a mobile application and a web application. The mobile application is designed for residents, with the primary goal of connecting the resident with neighboring units. The web application is for building administrators to verify and manage residents of a building. Although this project can be defined as a social network, it is not like most other social applications – social media postings are omitted. This project solely focuses on connecting residents together and is not focused on being a comparison game of likes, comments, and followers.

To make this project successful, the project is laid out in terms of planning, analyzing, designing, and developing, with a completion date of May 3, 2025. This document is the proposal for this project and is in the planning phase. A vague description in terms of what the project consists of has been developed, with more technical terminology in the high-level solution. Very technical terminology has been omitted. Basic functionality of the applications will be completed before any other features are pulled in to ensure the project can be completed with the success criteria met. Since this project is a large undertaking, there is an in-depth risk management plan to minimize issues occurring in this project. Proposed technologies will be researched, and proof of concepts will be developed before a final decision on the technology stack is made.

|  |
| --- |
| History and Signoff Sheet |

**Change Record**

|  |  |  |
| --- | --- | --- |
| **Date** | **Author** | **Revision Notes** |
| 10/11/2024 | Grace Radlund | Initial draft for review/discussion |
| 10/22/2024 | Grace Radlund | Updated document per instructor feedback |
| 4/8/2025 | Grace Radlund | Preparing document for showcase |
| 4/24/2025 | Grace Radlund | Preparing for project portfolio |

|  |
| --- |
| **Overall Instructor Feedback/Comments** |

|  |
| --- |
| **Overall Instructor Feedback/Comments** |

**Integrated Instructor Feedback into Project Documentation**

Yes  No

**TABLE OF CONTENTS**

Functional Requirements 5

Non-Functional Requirements 5

Technical Requirements 6

Logical System Design 7

User Interface Design 8

Reports Design 12

Functional Requirements

Refer to the Sprint Back Log spreadsheet. Each feature is broken down on separate sheets in the workbook.

Non-Functional Requirements

Refer to the Sprint Back Log spreadsheet. Listed as NFR in the workbook.

Technical Requirements

The following are tools, languages, frameworks, databases, and IDE’s that will used to successfully execute the project. Their versions and justifications for each are also listed.

|  |  |  |
| --- | --- | --- |
| **Technology or Tool** | **Version** | **Justification** |
| React Native | v0.74 | JavaScript framework used to develop the applications. |
| Expo | v51.0.0 | React Native framework used for easier development of applications. |
| NativeWind | v3.3.2 | CSS framework for universal styling of the applications. |
| Typescript | v5.6.3 | JavaScript superset that will be used for React Native. |
| Express | v4.2.1 | Node.js framework that will be used for building the API. |
| Node.js | v20.18.0 | Runtime environment that will execute the API. |
| Supabase | v1.204.4 | Cloud service provider for the PostgresSQl database. |
| supabase-js | v2.45.4 | JavaScript client that will interact with the PostgreSQL database. |
| PostgreSQL | v15.6 | Database that will used to store data. |
| Draw.io | v24.7.17 | Free tool for technical diagramming. |
| Expo Go | v2.31.6 | Free tool for local development builds. |
| Figma | v16.13.3 | Free tool for wireframing. |
| GitHub | v3.4.6 | Free tool for code management. |
| Jira | v9.17.0 | Free tool for project management. |
| MacBook Pro M1 chip | Ventura 13.5.2 | Computer used to execute the project. |
| Visual Studio Code | v1.91.4 | Free IDE for code development. |
| Word | v16.80 | Free tool for documentation. |

Logical System Design

Knockk is a full stack application project. Below is a diagram of how technologies will communicate with one another.

A diagram of a software application

Description automatically generated

The user will interact with the mobile application, while the admin will interact the web application. These applications will run in a sandboxed environment, utilizing Expo Go. Both frontend applications communicate to the REST API, which communicates with the relational database.

User Interface Design

The sitemap below displays the flow of both the resident and admin application. A key is provided for more information on meanings of different symbols, colors, etc.

A screenshot of a computer

Description automatically generated

The following wireframes are low fidelity designs for the admin application.

A screenshot of a computer

Description automatically generated

Login Page

A screenshot of a computer

Description automatically generated

Manage Residents Page

A screenshot of a computer

Description automatically generated

Verify a Resident Page

A screenshot of a computer

Description automatically generated

Activate a Resident Page

A screenshot of a computer

Description automatically generated

Error Page

Please refer to the “Knock Mobile Application Wireframes” zip for mobile application wireframes.

Reports Design

The admin application has two main reports that are generated within the application:

Managing residents

The admin user will be able to view residents. This report will include:

* User id
* First name
* Last name
* Email
* Floor number
* Room number
* Bedroom
* Lease start date
* Lease end date

The report will populate 10, 25, or 50 residents based on what the admin user chooses; default will be 25 residents. The user will have 5 visible pages of pagination. The admin user will be able to sort based on floor number increasing, floor number decreasing, last name a to z, or last name z to a; default will be floor number increasing. There will be no filtering on the data as this is an out-of-scope feature.

Verifying residents

The admin user must first verify the data that the user entered during registration. This report will only display one user and will include:

* User id
* First name
* Last name
* Email
* Floor number
* Room number
* Bedroom number

From this information, the admin user will either reject the resident or activate their account (takes a few extra steps). Verification will only verify one resident at a time, so there will be no sorting or filtering of information.

The user application has no reports.