*Software Requirements*

*Specification*

*for*

*LocalSquare*

Table of Contents

1. **Introduction**
   1. Document Purpose
   2. Intended Audience
   3. Product Scope
2. **Overall Description**
   1. Product Perspective
   2. Product Features
   3. Users
   4. Design and Implementation Constraints
   5. Assumptions and Dependencies
3. **System Features**
4. **External Interface Requirements**
5. **System Non-Functional Requirements**
6. **Other Requirements**

**1. Introduction**

**1.1 Document Purpose**

This document has been written for the purpose of guiding the design, development, and maintenance of the application. It will outline the project requirements, features, and implementation.

**1.2 Intended Audience**

The intended audience for this document are developers, who will need to build the application to the requirements and specifications defined, and the trainers who will need to understand the documentation and design process.

**1.3 Product Scope**

The application is a group and community-centered social media platform. Its purpose is to provide a space for groups and communities to connect and share information. It is designed to be modular and fit any group or community's needs. The groups created on the platform are also relational, supporting connections with other groups, letting users traverse connected groups to find new communities and connections.

Unlike traditional social media applications, groups are the building blocks of this application, rather than users. Groups are made up of tabs, which outline the interactions users can have with the group and other users. For example, a group could have a mix of forums, group information, and chats, or only one of the examples. Users can define their profile on a per-group basis, and set group-related personal info, such as their role in the group.

Group admins can define which interactions the users can make, and set special conditions for user interactions. For example, a user making a post, or setting their role, could require admin approval before the change goes live.

This application aims to bring groups of people closer together in a healthy way. The application does not rely on advertising or maintaining user retention. A good case study for this application is ‘Discord’, while Discord focuses on online groups, this application focuses more on physically located groups, and supports more ways of sharing information, rather than focusing on chat rooms.

**2. Overall Description**

**2.1 Product Perspective**

The application will be hosted with AWS infrastructure, relying on their expansive services to run the servers, and databases, and serve the front-end content. The servers will be elastic and scalable to handle variable loads.

The application will be built with Node.js and Express for the back-end, MongoDB for the database, and React for a responsive and modular front-end.

The application will first be a web app with support for adding mobile applications in the future.

**2.2 Product Features**

User and authentication - create an account and get authenticated.

Find groups - search for groups based off filters and search parameters.

Join groups - request to join groups / accept user join requests.

Group page - browse and interact with group tabs / edit group page content and settings.

Group user profile - set user profile for group based off existing profile data + custom data specific for that group.

**2.3 Users**

On a group level, users are split into two main classes, default users, and administrators.

Default users are users who interact with the content and tabs of a group for their own sake. They are there to participate in the group, consuming and posting content and information.

Administrators are there to set permissions, accept requests for actions users have made which require approval, and moderate the interactions users make within the group. Admins can have different levels of roles within the group, and permissions to do different things.

**2.4 Design and Implementation Constraints**

Variable range in network speeds of users should be taken into account in relation to content being served to users (such as images and large files).

**2.5 Assumptions and Dependencies**

The application will rely on AWS services for hosting the full stack of the application. The application will rely on MongoDB for storing the data of all the groups and users.

The application will rely on Node.js and Express for the back-end to complete all the user actions.

The application will rely on React for the front-end.

The application will rely on Socket.io for the client-server bidirectional communication.

**3. System Features**

**3.1 User Login/Signup and Authentication**

The ap

**3.2 User Main Profile**

**3.3 Group Search**

**3.4 Previewing Groups**

**3.5 Joining Groups**

**3.6 Navigating Between Groups**

**3.7 Creating Groups**

**3.8 Interacting With Groups**

**3.9 User Group Profile**

**3.10 Moderating Groups**

**3.11 Traversing Groups**