*Wayfarer*

Requirements Document

Last Revised: October 30, 2020

# Griffin Corey, Zachary Fox, Thomas Saulnier, Tapan Nepal

# Introduction

There is a community of people who enjoy the challenge of visiting every town in a particular state or region. The intent is primarily for recreation. Specifically: individuals, couples, friends, or families might seek new experiences by visiting towns to see sights, shop, eat at restaurants, attend events, or just to say they have been there. For the sake of simplicity, we will call the people (individuals or groups) who do this “visitors.”

Wayfarer is a system that allows visitors to manage their exploration of the towns of Vermont. It gives visitors a way to record information about each visit, in a journal-like manner, and share that information with others. Ultimately, Wayfarer is intended to support members of the 251 Club of Vermont by connecting to the 251 Club’s backend database of users and recorded comments and photos. However, as a simplification, Wayfarer will initially be designed for use by anyone and have no explicit connection with the 251 Club. Although the concept of “town bagging” is not unique to Vermont, Wayfarer as currently imagined focuses on that one state, also as a simplification.

The rest of this document describes the detailed functional and non-function requirements for Wayfarer.

# Functional Requirements

The Wayfarer system shall only support visitors who are visiting towns in the state of Vermont. Wayfarer shall not attempt to interact with the existing 251 Club web site. However, see the Expected Extensions non-functional requirement. These limitations are likely to be lifted in the future.

## Actors

The following actors (roles) shall be supported:

1. **Visitor**. An individual (or a group, but treated as an individual by the system) who wishes to look at information entered by a **Member** of the 251 club.
2. **Member.** An individual (or a group, but treated as an individual by the system) who is a dues paying member of the Vermont 251, this user wishes to record information about the town they visit, create posts and comment on other users.
3. **Admin.** An individual (or a group, but treated as an individual by the system) who has write privilege to the Lemuria database. Admins have Member privileges as well.

## Use Cases

|  |  |
| --- | --- |
| **ID:** | **Write-Post** |
| **Title:** | Create Post Artifact |
| **Description:** | A member creates a post related to one town that the visitor has visited. |
| **Primary Actor:** | Member, Admin |
| **Preconditions:** | The member must be authenticated to the system |
| **Postconditions:** | The comment provided by the Member has been recorded in the artifact database |
| **Main  Success Scenario:** | * Member enters the town * Member enters the date * Member writes/edits the post * Member explicitly saves post |
| **Extensions:** | [TODO: What happens if there is no network connectivity?]  If no network connection, save as draft (.txt) on client side. |
| **Frequency of Use:** | Frequently. |
| **Status:** | Not Implemented |
| **Owner:** | Griffin Corey |
| **Priority:** | Very high. This is essential functionality. |

|  |  |
| --- | --- |
| **ID:** | **View-Post** |
| **Title:** | Pull Post Artifact |
| **Description:** | All users (visitor, member, admin) can view posts written for a given city. |
| **Primary Actor:** | Visitor, Member, Admin |
| **Preconditions:** | Any user can access this information, so the only precondition is to have navigated to the area in question |
| **Postconditions:** | The post requested is displayed on the user’s screen for viewing |
| **Main**  **Success Scenario:** | * User selects the town * User selects a date for posts * User obtains a list of posts matching these criteria |
| **Extensions:** | [TODO: What happens if there is no network connectivity?]  If no network connection, will return unable to access database at this time. |
| **Frequency of Use:** | Frequently. |
| **Status:** | Not Implemented |
| **Owner:** | Griffin Corey |
| **Priority:** | Very high. This is essential functionality. |

|  |  |
| --- | --- |
| **ID:** | **Delete-Post** |
| **Title:** | Delete Post Artifact |
| **Description:** | All members can delete posts made by themselves. Admin can delete any post they want. |
| **Primary Actor:** | Member, Admin |
| **Preconditions:** | For the member it must be a post made by them. Admin has no precondition it just needs the post to delete. |
| **Postconditions:** | The post is deleted from the database. |
| **Main**  **Success Scenario:** | * User selects the post to delete and the post is removed from the database. |
| **Extensions:** | [TODO: What happens if there is no network connectivity?]  If no network connection, will return unable to access database at this time. |
| **Frequency of Use:** | Frequently. |
| **Status:** | Not Implemented |
| **Owner:** | Griffin Corey |
| **Priority:** | High. This is essential functionality. |

|  |  |
| --- | --- |
| **ID:** | **Create-Account** |
| **Title:** | Create Account Artifact |
| **Description:** | Asks for Member Name, username and password. Must enter password twice. Provides link to: <https://www.vt251.com/vt/online_member_form.php> .  Our program will assume they registered online and paid and so will give them member privileges. Assigns each new user member id. |
| **Primary Actor:** | Visitor, Admin |
| **Preconditions:** | Any unsigned in user can create a new account. You must have connection to do this. |
| **Postconditions:** | The new account sends back either successful account creation or unsuccessful account creation. |
| **Main**  **Success Scenario:** | * Visitor successfully creates a new account and can proceed to Login to start posting |
| **Extensions:** | [TODO: What happens if there is no network connectivity?]  If no network connection, will return unsuccessful account creation |
| **Frequency of Use:** | Frequently. |
| **Status:** | Not Implemented |
| **Owner:** | [Thomas Saulnier] |
| **Priority:** | Very high. This is essential functionality. |

|  |  |
| --- | --- |
| **ID:** | **Login** |
| **Title:** | Authenticate User As Member |
| **Description:** | A visitor can log into the program as a member if they previously created an account. |
| **Primary Actor:** | Visitor |
| **Preconditions:** | Any user can attempt to log in, but only those with valid credentials will be able to proceed as a member. |
| **Postconditions:** | The user is validated as a member and obtains the relevant privileges in the program. |
| **Main**  **Success Scenario:** | * User navigates to the log in area * User inputs their member credentials * User is signed in as a member |
| **Extensions:** | If no network connection, will return unable to access database at this time. |
| **Frequency of Use:** | Frequently. |
| **Status:** | Not Implemented |
| **Owner:** | [Thomas Saulnier] |
| **Priority:** | Very high. This is essential functionality. |

|  |  |
| --- | --- |
| **ID:** | **Edit-Profile** |
| **Title:** | Edit information on member profile |
| **Description:** | A member or admin can edit their information for their profile, such as their name, username, or password. Note: the profile ID cannot be changed. |
| **Primary Actor:** | Member, Admin |
| **Preconditions:** | The user must already have created a profile to make changes to theirs. |
| **Postconditions:** | The user now has new profile information, except their ID which cannot be changed. |
| **Main**  **Success Scenario:** | * User requests to edit profile * User inputs new information * The new information is saved |
| **Extensions:** | If no network connection, will return unable to access database at this time. |
| **Frequency of Use:** | Frequently. |
| **Status:** | Not Implemented |
| **Owner:** | [Thomas Saulnier] |
| **Priority:** | Medium. A member does not need to change their profile information/credentials but may wish to do so. |

|  |  |
| --- | --- |
| **ID:** | **Change-Member-Status** |
| **Title:** | Changing Member Status |
| **Description:** | Admin can update a member in the database from being a member or not. |
| **Primary Actor:** | Admin |
| **Preconditions:** | Admin privileges |
| **Postconditions:** | The user is validated as either a member or a non-member and their role is updated. |
| **Main**  **Success Scenario:** | * Admin verifies that the user being updated is either a member or non-member. * The member’s status is updated in the database. |
| **Extensions:** | If no network connection, will return unable to access database at this time. |
| **Frequency of Use:** | Frequently. |
| **Status:** | Not Implemented |
| **Owner:** | [Thomas Saulnier] |
| **Priority:** | Very high. This is essential functionality. |

|  |  |
| --- | --- |
| **ID:** | **Edit-Table** |
| **Title:** | Edit the SQL tables. |
| **Description:** | An admin can edit the SQL tables and add another row/column or change functionality of the table. |
| **Primary Actor:** | Admin |
| **Preconditions:** | Be an Admin. |
| **Postconditions:** | New posts must follow this layout. |
| **Main**  **Success Scenario:** | * The table is edited accordingly |
| **Extensions:** | You must be logged in as admin which requires access to database. |
| **Frequency of Use:** | Frequently. |
| **Status:** | Not Implemented |
| **Owner:** | Griffin Corey |
| **Priority:** | Very high. This is essential functionality. |

# Database setup

USERS

Uname – self explanatory VARCHAR 32

Passwd – self explanatory VARCHAR 32

Uid – user id int 8

Ismember – is a payed member Boolean

DATA

Uid – self explanatory int 8

Date – self explanatory DATETIME

Comment – spot for user to describe their visit VARCHAR 255

Locationid – identifier for location (town) int 8

LOCATION

Locationid – identifier for location int 8

Location – actual name of lactation VARCHAR 255

# Non-Functional Requirements

## Platform

* **Platform-Architecture**. Wayfarer shall be implemented as a client/server system with a client program used by visitors and a server backend that stores all recorded artifacts. This architecture anticipates a future extension whereby Wayfarer (optionally) interacts with the backend systems of the 251 Club of Vermont or possibly other, similar organizations in other states.
* **Platform-Multiuser**. Wayfarer shall support multiple users of the same backend system. This also anticipates a future extension supporting the 251 Club of Vermont (or similar).
* **Platform-Client**. The client application which visitors use shall be Linux emulator.
* **Platform-Backend**. The backend system shall execute on SQL.

## Performance

## The minimum number of resources needed for this project are git hub and Lemuria.

## Security

The information stored in the database is not very sensitive. The user profiles contain only a username and password to manage their entries and whether they are a member or not. Payment processing for members is done outside of this software and as such we needn’t worry about credit card information or money transfers. Ergo, a simple SHA2 unsalted hash function for password storage should suffice since the fallout from an attack would be very minimal.

## User Characteristics

## The users should be technically capable enough to use Linux in its basic functions and should be able to use commands competently.

## Scale

The system is designed to support as many users and their contents depending on the Lemuria database.

## Documentation

Documentation for the various roles, as well as the login/authentication cases and database management cases, should be included. Actions like editing a profile and adding a post are fairly straightforward and should need less detailed documentation if any.

## Data Formats

The information for various points of interest will be stored in plaintext, and so will usernames, but the passwords will be hashed and stored as plaintext hashes. All the required information will be stored as plaintext in the SQL database on Lemuria.

## Internationalization

Our project does not support multiple languages or cultures, it only supports the 251 club of Vermont and what they want.

## Expected Enhancements

* **Enhancement-251-Backend**. Wayfarer will likely need to be enhanced to (optionally) interact with the back-end databases of the 251 Club of Vermont. Specifically, it should eventually be possible for a visitor who is a member of the club to use his/her club credentials to log in and then upload artifacts to the club’s server (for presentation on the club web page), and also to share artifacts with, and view artifacts shared by other club members.
* **Enhancement-Multistate**. Wayfarer may need to be extended to support states other than Vermont.

## Date

* **Date-Presentation**. A presentation of the system shall be given at the end of the Fall 2020 semester.