

Software Requirements Specification

for

OverLunch

Document Version 0.9

Prepared by

Greg Carrico

Benjamin Groman

Shaun Godfrey

SUNY Oswego

12/5/2020

Introduction	3
Purpose	3
Product Scope	3
Overall Description	3
Product Overview	3
Goals	3
Functionators	4
Expectors	4
Design and Implementation Constraints	5
Appendix A: Glossary	6
Appendix B: Scenarios	8
S1.0 Initial Setup	8
S2.0 View/Change Account Settings	8
S3.0 HELP!	8
S4.0 Browsing events	8
S4.1 Sort Events	9
S4.2 Filter Events	9
S5.0 Select an event/ view meeting details	9
S5.1 Join an event: Interested	9
S5.2 Join an event: Confirmed	9
S6.0 Invite non-user to join a meeting	10
S7.0 View User Profile	10
S7.1 Invite User to Friends List	10
S7.2 Block a user	10
S8.0 Cancel an event	11
S9.0 Create an Event	11

Introduction

This SRS is a living document, and may be modified with the consent of the majority of the development team. A record of all changes made to this document can be found in the Revision History of the OverLunch project on [Github](https://github.com/gcar42/OverLunch.git). < <https://github.com/gcar42/OverLunch.git>>

Purpose

The purpose of this document is to specify the design, development, implementation, and testing of an application intended to aid SUNY Oswego students in finding a lunch companion on little to no advanced notice. This document will specify system requirements, goals, scenarios, and solutions, and will list qualities, functional requirements, and constraints of the system under development. This document will also describe quality assurance measures and testing procedures.

Product Scope

This product should include a user interface that allows users to browse, create, and view detailed information about events. The user interface should also support related functions as outlined in section 2.2 Goals. Providing menus (of available food) is outside the scope of this project. This product should make a reasonable effort to identify the times of operation for a given venue, but makes no claims as to the accuracy of this information.

Overall Description

Product Overview

The OverLunch application is intended to aid SUNY Oswego students in finding a lunch companion on little to no advanced notice. Users of the app are primarily SUNY Oswego students, but anyone with a @oswego.edu email address may create an account and use the application. The application is accessible via a web interface.

Goals

The goals enumerated in this section have been derived or taken directly from documents, conversations, and various communications between the OverLunch development team and the other stakeholders, such as potential users of the OverLunch app. These goals are categorized in three groups, functionators, expectors, and excitors, defined thus:

Functionators are goals that represent the development of a minimally viable product. A product developed with only these goals would not be expected to satisfy the objectives of the OverLunch project. Rather these goals should be viewed as elemental components of a more complete and desirable product.

Expectors are goals that represent the expected properties of the completed product under development. If implemented, these goals are likely to lead to a satisfying final product that achieves the OverLunch Project's desired core functionality.

Excitors are goals that might be less obvious, but add value and possibly a "wow" factor to the final product.

Functionators

GF1 | A user should be able to create an event.

GF2 | A user should be able to browse events.

GF3 | A user should be able to join an event.

GF3.1 | A user should be able to join an event, interested.

GF3.2 | A user should be able to join an event, confirmed.

GF4 | A user should be able to cancel an event.

GF5 | A user should be able to remove self from an event.

GF6 | A User should be able to see detailed information about an event.

GF6.1 | Event details should include users who have tagged the event as
“interested.”

GF6.2 | Event details should include users who have confirmed they will attend
the event.

GF6.3 | Event details for unconfirmed events should include the time range in
which the event could occur.

GF6.4 | Event details should include the name of the venue selected for the
event.

Expectors

GE1 | The OverLunch app should allow users to request permission of other users to add them
to a friends list.

GE2 | A user should be able to accept a request to be added to a friends list.

GE2.1 | A user should be able to decline a request to be added to a friends list.

GE2.3 | Friendship is a mutual relationship. (It applies to both parties. If I grant a
friendship permission, the requestor becomes my friend).

GE3 | A user should be able to sort events.

GE4 | A user should be able to filter events.

GE5 | A user should be able to block other users.

GE5.1 | A blocked user should not be able to view the blocking user’s profile.

GE5.2 | A blocked user should not be able to see events created by the blocking user.

GE6 | A user should be able to set default event parameters.

GE6.1 | Default event parameters should include venue.

GE6.2 | Default event parameters should include a time range.

Design and Implementation Constraints

Limitations to the design and development of OverLunch include the following:

C1 | The OverLunch app shall restrict the time range in which events may be scheduled to

Today and Tomorrow.

C2 | The OverLunch app shall not retain a record of events or related data outside the range of

Today and Tomorrow.

C3 | Users of the OverLunch app must have an @oswego.edu email address.

C4 | The OverLunch app should not permit events to be scheduled outside of a chosen venue's

hours of operation, as provided by the venue.

C5 | The OverLunch Project shall be completed by December 11, 2020.

C6 | The OverLunch app shall take reasonable steps to protect user data.

C6.1 | Reasonable steps shall be determined by the agreement of the majority of the development team.

Appendix A: Glossary

Block (a user)(v)

To block a user is to deny the blocked user the ability to see or join events created by the user who initiates the block. Blocking is symmetric, so the user initiating may no longer view or join events created by the blocked user.

Cancel (an event)(v)

A user canceling an event is removed from that event. If neither the host nor any other confirmed user remains, the event is deleted.

Confirmed (adj)

1. The status of a user in relation to an event, indicating that the user intends to be present at the given event. where a user who has committed to attend an event is considered Confirmed for that event.
2. The status of an event, indicating that at least one user is confirmed to attend.

Event (n)

An event is the specification of a time frame and a venue by a user for either a potential (interested) or confirmed lunch meeting.

Filter (n)

A filter is a limitation placed on the events displayed in a list. For example, a user might select a specific venue as a filter in order to restrict his displayed events to those occurring at that venue.

Friend (n)

A Friend is an OverLunch user who has accepted another user's friendship request. Friends can view each other's profiles, which show all of the events that the friend has either confirmed or is interested in.

Interested (adj)

The status of a user in relation to a meeting, indicating that the user might attend the event.

Meeting (n)

Synonymous with "event".

Sort(v)

To arrange the events in a list according to an order of preference.

User Profile (n)

1. Specific information about a user, including his Screen Name, Username, and any events to which the user is either confirmed or interested.

2. The user interface screen displaying user profile information. Note that the user profile screen displays a different set of information to the profile's owner than to other users.

Username

This is the unique name that OverLunch uses to identify a user. It consists of the prefix of the user's oswego.edu email address, and it may not be changed by the user.

Screen Name

This is a name chosen by the user which identifies them to other OverLunch users. It is the name that will be displayed when the user joins an event, and it may be changed at the user's discretion.

Appendix B: Scenarios

S1.0 Initial Setup

Matt receives an invitation via email to use OverLunch from his classmate Charles. Charles had already told Matt about the app, so Matt clicks the link.

A browser window opens prompting Matt to enter his Oswego.edu email address and select either sign-in or sign-up.

He enters his email address and selects the sign-up option. He is prompted to enter a Screen Name (which is what other users will see) and to create a password for the OverLunch app. The display tells him that an email confirmation link has been sent to his address.

The email will contain a link to start using the OverLunch app.

S2.0 View/Change Account Settings

Matt notices that he included a typo in his screen name. Hoping he won't have to jump through hoops to change it, he clicks the gear icon on the home screen to open the Settings options.

One of the items on the Settings page is User Account. That looks promising, so he clicks it.

The Account Setting page that opens shows him the email address he signed up with, and his Screen Name. He is pleased to discover he only has to click his screen name and retype it in order to change it.

S3.0 HELP!

Matt closes the User Account Settings window and spots another item on the Settings screen that interests him: "Help".

(Option 1) He clicks the button, causing a browser window to open to a page displaying the concise, yet exquisitely written OverLunch user manual.

(Option 2) He clicks the button, causing a new OverLunch screen to open, displaying the concise, yet exquisitely written OverLunch user manual.

Overcome by his genetic propensity for toxically masculine behavior, he decides he can figure it out on his own, and closes the browser window.

S4.0 Browsing events

Matt clicks the link in his confirmation email and is taken to the OverLunch Login screen.

After entering his email address in the Email field and password in the password field, Matt clicks the login button and is shown the Overlunch home screen.

The home screen shows a list of cards representing the lunch events of other OverLunch users.

He notices that the events occurring sooner are on the top of the list. (default sort, no filters).

Aside from the event tiles, he sees 2 obvious buttons. One is a plus sign that hovers near the bottom of the screen (New Event), and the other is a funnel-shaped icon near the top-right of the screen (Filter and Sort options).

S4.1 Sort Events

Clicking the funnel, Matt toggles the sort-order of the list of events to show the most recent events at the venues closest to his location.

S4.2 Filter Events

Matt scrolls through the list of events noticing that near the end of the list the cards are...

(a different color? Some visual cue to identify that these are for tomorrow, not today).

He is interested in meeting someone for lunch today, so he clicks the funnel icon again, and selects the filter option to show only Today's events.

The list updates (keeping his sort options) and tomorrow's events are no longer in the displayed list.

S5.0 Select an event/ view meeting details

Matt selects the 1st event in the list, which is the one that will be occurring soonest, and a window showing more details opens.

The "Detail" view shows that SCREEN_NAME_ONE and SCREEN_NAME_TWO have expressed interest in this meeting at (VENUE), which could begin as soon as 11:30 and end as late 1:45 (Event Window).

S5.1 Join an event: Interested

This all sounds good to Matt. He still wants to look at other events, so he selects "I'm Interested" instead of the "I'll Be There" option, adding his name to the list with the other two users.

S5.2 Join an event: Confirmed

Shortly after Matt joins the event, the bar at the top of the event changes color and displays the word "Confirmed". SCREEN_NAME_ONE has confirmed that (s)he will be there.

S6.0 Invite non-user to join a meeting

Matt thinks that his classmate Lindsey would probably like to join this meeting. He closes the Detail view (because the option to share should not be hidden) and opens the context menu for the event he just confirmed.

One of the options in this menu is "Invite a guest to this meeting." He selects that option and types Lindsey's email address. In the optional Message to Guest field he types "Hi Lin. It's Matt. Join me and Chuck for lunch!", and then clicks the INVITE button.

(OverLunch should now check for Lindsey's email address in the user database. Not finding it means Lindsey will be sent a signup link. Perhaps it can include a token that automatically shows this meeting once her signup is complete.)

If Lindsey is a user, she will be sent a meeting invitation link via email, but when she looks at the Events list, it should be immediately obvious that she has been invited.

S7.0 View User Profile

With a few minutes to spare between classes, Matt explores some of the OverLunch meeting details. Because he and Charles are OverLunch Friends, Matt can see more information about Charles by viewing Charles' user profile.

He discovers this by clicking Charles' name in the meeting they have both already confirmed. Charles' profile shows the other events that he has either confirmed or marked "Interested".

(Viewing your own profile page shows your username, email address, and all events occurring either today or tomorrow to which you are either confirmed or interested.)

S7.1 Invite User to Friends List

Matt selects one of the events that Charles is Interested in. He sees that Charles' sister Bonnie has been confirmed for the event.

Matt selects Bonnie's name to see her profile, and gets a message saying "You may only view user profiles of people in your Friends list. Would you like to invite Bonnie to your Friends list?" Matt quickly selects YES.

S7.2 Block a user

Later, after Bonnie has accepted Matt's invitation to be friends, Matt notices that Bonnie's ex-boyfriend Raphael has confirmed for tomorrow's meeting and that Bonnie is no longer in the meeting's list. She must have cancelled.

Matt is about to select the Cancel option from the meeting's context menu, but first, he selects Raphael's name and sees an option to Block User, which he selects.

A message is displayed that explains Blocking a user works in both directions -- Matt will no longer see meetings that Raphael has created, and Raphael will no longer see any meeting created by Matt. They might still see each other attached to meetings created by other users.

Matt selects the Confirm option. Raphael seemed like a decent guy, but it could be awkward if he wanted to join meetings with Bonnie and had just had lunch with Raphael yesterday.

S8.0 Cancel an event

(Remove yourself from an event. It will still exist for other attached users).

Still looking at the details of the meeting with Raphael, Matt returns to the context menu and selects "Cancel". The confirmation message says "You will be removed from this meeting. Continue?". Matt chooses yes, and the meeting detail screen closes.

S9.0 Create an Event

After enjoying a lunch meeting scheduled on OverLunch, Matt wants to do it again tomorrow. He opens the app on his phone, opens the OverLunch home screen and clicks the green "Plus" button to create a new event.

He selects VENUE_X. (From a dropdown menu? From a list of venues that opens in a new window?)

Once the venue is chosen, Matt can choose when he wants his event to occur. (Dev perspective: The app will use the given hours of operation for the venue to prevent a meeting being scheduled outside that time window).

He selects 12:30 as the earliest time he could start a lunch event, and 2:30 as the latest time the event could end.

A summary window tells him "You are creating a new event for tomorrow between 12:30 and 2:30 at VENUE_X. |ADD it| |EDIT it| |CANCEL it|

Matt selects "ADD it", creating an unconfirmed event for tomorrow.