



California State University, Long Beach  
College of Engineering  
Computer Engineering & Computer Science  
CECS 491A, Section 07  
Professor M. Bactor  
Spring 2019

### **Team PhuketList**

<https://stevengarciadev.github.io/PhuketList/>

Richie Espinosa-Garcia, Steven Garcia, Sukrit Mehra, Kenny Nguyen

## **Abstract**

## Table of Contents

<b>Abstract</b>	<b>2</b>
<b>Table of Contents</b>	<b>3</b>
<b>1. Introductions and Background</b>	<b>6</b>
1.1. Statement of Problem Area (brief, nontechnical)	6
1.2. Previous and Current Work, Methods and Procedures (representative)	6
1.3. Background	6
1.4. Brief Project Description (overview of new, extended or different functions, structure or operation)	6
1.5. Purpose/Objectives/justification of Project (theoretical, practical, or educational impacts on users)	6
1.6. Teamwork assignment and accomplished	6
<b>2. System Functional Specification</b>	<b>7</b>
2.1. Functions Performed (itemize and describe)	7
2.2. User Interface Design	7
2.3. User Input Preview	7
2.4. User Output Preview	7
2.5. System Database/File Structure Preview	7
2.6. External and Internal Limitations and Restrictions	7
2.7. User Interface Specification	7
2.7.1. Interface Metaphor Model	7
2.7.2. User Screens/Dialog	7
2.7.3. Report Formats/Sample Data	7
2.7.4. Online Help Material	7
2.7.5. Error Conditions and System Messages	7
2.7.6. Control Functions	7
<b>3. System Performance Requirements</b>	<b>8</b>
3.1. Efficiency	8
3.2. Reliability	8
3.2.1. Description of Reliability Measures (accuracy, precision, consistency, reproducibility, etc.)	8
3.2.2. Error/Failure Detection and Recovery (failure modes, failure consequences, error logging and reporting, manual and automatic recovery procedures)	8
3.2.3. Allowable/Acceptable Error/Failure Rate	8
3.3. Security	8

3.3.1. Hardware Security	8
3.3.2. Software Security	8
3.3.3. Data Security	8
3.3.4. Execution Security (user validation)	8
3.4. Maintainability	8
3.5. Modifiability	8
3.6. Portability	8
3.7. Others	8
<b>4. System Design Overview</b>	<b>9</b>
4.1. System Data Flow Diagrams	9
4.2. System Structure Charts	9
4.3. System Data Dictionary	9
4.4. System Internal Data Structure Preview	9
4.5. Description of System Operation (high level)	9
4.6. Equipment Configuration (diagram and description)	9
4.7. Implementation Languages (which and why)	9
4.8. Required Support Software (preexisting)	9
<b>5. System Data Structure Specifications</b>	<b>10</b>
5.1. Other User Input Specification	10
5.1.1. Identification of Input Data	10
5.1.2. Source of Input Data (NOT input device)	10
5.1.3. Input Medium and/or Device	10
5.1.4. Data Format/Syntax	10
5.1.5. Legal Value Specification	10
5.1.6. Examples	10
5.2. Other User Output Specification	10
5.2.1. Identification of Output Data	10
5.2.2. Destination of Output Data (NOT output device)	10
5.2.3. Output Medium and/or Device	10
5.2.4. Output Format/Syntax	10
5.2.5. Output Interpretation (meaning of output)	10
5.2.6. Examples	10
5.3. System Database/File Structure Specification	10
5.3.1. Identification of Database/Files	10
5.3.2. (Sub)systems Accessing the Database (creating, updating, using, frequency)	10

5.3.3. Logical File Structure (record formats, file organization, access methods, rationale, examples)	10
5.3.4. Physical File Structure (storage device, blocking, organization, access, etc.)	10
5.3.5. Database Management Subsystems Used (internal or external)	11
5.3.6. Database Creation and Update Procedure (if NOT by system)	11
5.4. System Internal Data Structure Specification	11
5.4.1. Identification of Data Structures	11
5.4.2. Modules Accessing Structures (creating, updating, using)	11
5.4.3. Logical Structure of Data (format, organization, access, rationale, examples)	11
<b>6. Module Design Specifications (for each module)</b>	<b>12</b>
6.1. Module Functional specification	12
6.1.1. Functions Performed	12
6.1.2. Module Interface Specifications (input/output arguments/global variables/files)	12
6.1.3. Module Limitations and Restrictions	12
6.2. Module operational Specification	12
6.2.1. Locally Declared Data Specifications (variable dictionary)	12
6.2.2. Algorithm Specification (flowchart, pseudocode, decision table, etc)	12
6.2.3. Description of Module Operation	12
<b>7. System Verification</b>	<b>12</b>
7.1. Items/Functions to be Tested	13
7.2. Description of Test Cases	13
7.3. Justification of Test Cases	13
7.4. Test Run Procedures and Results	13
7.5. Discussion of Test Result	13
<b>8. Conclusions</b>	<b>14</b>
8.1. Summary	14
8.2. Problems Encountered and Solved	14
8.3. Suggestions for Better Approaches to Problem/Project	14
8.4. Suggestions for Future Extensions to Project	14

# **1. Introductions and Background**

## **1.1. Statement of Problem Area (brief, nontechnical)**

The purpose of this project is to create a social network that provides its users a place to share each other's bucket list items. The network should provide users with features that provide a different array of social interactions online. Users should be able to friend each other and view each other's content. Users are to be able to interact with other strangers that also share similar interests based off each other's bucket list items.

## **1.2. Previous and Current Work, Methods and Procedures**

### **1.2.1. Previous/Current Work**

All of our work is listed on our team website:

<https://stevegarcia.dev/github.io/PhuketList/#progress-reports>

Work provided: Progress Reports Week 1-14, Labs 1-4

### **1.2.1. Methods**

### **1.2.1. Procedures**

## **1.3. Background**

## **1.4. Brief Project Description (overview of new, extended or different functions, structure or operation)**

## **1.5. Purpose/Objectives/justification of Project (theoretical, practical, or educational impacts on users)**

## **1.6. Teamwork assignment and accomplished**



## **2. System Functional Specification**

- 2.1. Functions Performed (itemize and describe)
- 2.2. User Interface Design
- 2.3. User Input Preview
- 2.4. User Output Preview
- 2.5. System Database/File Structure Preview
- 2.6. External and Internal Limitations and Restrictions
- 2.7. User Interface Specification
  - 2.7.1. Interface Metaphor Model
  - 2.7.2. User Screens/Dialog
  - 2.7.3. Report Formats/Sample Data
  - 2.7.4. Online Help Material
  - 2.7.5. Error Conditions and System Messages
  - 2.7.6. Control Functions



### **3. System Performance Requirements**

#### 3.1. Efficiency

#### 3.2. Reliability

3.2.1. Description of Reliability Measures (accuracy, precision, consistency, reproducibility, etc.)

3.2.2. Error/Failure Detection and Recovery (failure modes, failure consequences, error logging and reporting, manual and automatic recovery procedures)

3.2.3. Allowable/Acceptable Error/Failure Rate

#### 3.3. Security

3.3.1. Hardware Security

3.3.2. Software Security

3.3.3. Data Security

3.3.4. Execution Security (user validation)

#### 3.4. Maintainability

#### 3.5. Modifiability

#### 3.6. Portability

#### 3.7. Others

## **4. System Design Overview**

4.1. System Data Flow Diagrams

4.2. System Structure Charts

4.3. System Data Dictionary

4.4. System Internal Data Structure Preview

4.5. Description of System Operation (high level)

4.6. Equipment Configuration (diagram and description)

4.7. Implementation Languages (which and why)

4.8. Required Support Software (preexisting)

## 5. System Data Structure Specifications

### 5.1. Other User Input Specification

#### 5.1.1. Identification of Input Data

One type of data that our application takes as input are string literals. This type of input is received whenever the user tries to key in a description or some kind of information.

Another input type that our application takes are image files when the user uploads a file.

Our application also listens to event inputs for UI related interactions like buttons. For more details on how these kind of inputs are being used, see 5.1.6.

#### 5.1.2. Source of Input Data (NOT input device)

The source of where the application would be from its users, mainly the customers that will be using this application to connect with other users in completing their task list.

Additionally Admins and other software developers can also be sources of input since they will be using this application to manage it, test it, and add additional features.

#### 5.1.3. Input Medium and/or Device

Users of our application would be able to interact with our application through the use of a mouse and a keyboard. The mouse is needed to interact with buttons and other UI elements. The keyboard is needed when users need to key in information, see 5.1.6 for more details.

#### 5.1.4. Data Format/Syntax

#### 5.1.5. Legal Value Specification

#### 5.1.6. Examples

Examples of where our application takes in user input are the following:

- searching through a user's task items using a search bar
- changing or creating a task item which involves naming the task
- changing a user's name and credentials
- writing descriptions for a post
- Commenting on a post
- Writing a description when reporting a post

- Writing in user's biography
- finding friends through a search bar
- registering users
- authenticating users through sign in
- Uploading an image either for user profile or a post via buttons
- Clicking buttons on navigation bar or hyperlinks to get to each page of the application
- Clicking the post button to submit a post users have made
- Clicking on buttons on a post to either like it, comment on it, or report it
- Clicking the add task button to create a task
- Clicking buttons on the task item to edit the task, delete it, or to mark it as complete
- Clicking on the dropdowns for filtering the order of task items
- Clicking the dropdown for the connect button or account button on nav bar to select even more options
- Clicking buttons in user settings to enable the editing of the user's name, email, and to set user's active or private states

## 5.2. Other User Output Specification

### 5.2.1. Identification of Output Data

The output of the application is its graphical view of the application that users can see. This display allows users to see their personalized information and others such as their task items. Also displayed is all the UI elements that the user can interact with to change the user's information and therefore the way the application is viewed, such as creating a task item will result in a new task item appearing on the screen.

### 5.2.2. Destination of Output Data (NOT output device)

The output will be delivered onto the frontend side of the application from the backend side. Each time a page is loaded, the frontend make requests to the API to grab the information associated with the user and the page from database. When retrieved, the information gets displayed for the user as output on the front end side.

### 5.2.3. Output Medium and/or Device

Computer monitors are the main medium needed for the output to be displayed. It is only through the monitor can users be able to view the state of the application and be able to interact with it.

#### 5.2.4. Output Format/Syntax

When the application gets displayed, what remains constant throughout each page is the navigation bar at the top that contains buttons that are meant to redirect users to the different pages in the application. There are fewer buttons here when logged out compared to when logging in since the application's features are only accessible to users who have an account. Each page has its own unique display, other than the navigation bar. The Mylist page contains list of the user's task items that the user can edit, a search bar to search through the tasks, and a text field that the user can use to create a new task. The task group page lists all the post that members of that page has posted, with users being able to interact with them. The activity page contains a list of post that the user has made, along with links of their task groups on the left. The edit profile page displays the user's profile image, biography, and bucketlist, with each in its own panel, while my profile page displays them without any editing features. Friends page has two different displays. If find friends button is clicked, it will display a search bar to find and add users while listing recommended users to add at the bottom of the search bar. If edit friends list button is clicked, it will display the user's current friends and buttons next to them that let you remove them. The message page, which is still a work in progress, has a start new group button which enables users to find other users to chat with, a text field at the bottom that lets users message users, and large box above it at the right that displays all of the messages sent. The user settings lists the user's name, email, active state, and private state, with a button on each so the user can edit these settings. The contact page, although as of now empty, is meant to display user's contact information. Both the sign in and register pages have a list of fields that the user uses to enter their credentials to either login or to build their account.

#### 5.2.5. Output Interpretation (meaning of output)

#### 5.2.6. Examples

### 5.3. System Database/File Structure Specification

#### 5.3.1. Identification of Database/Files

#### 5.3.2. (Sub)systems Accessing the Database (creating, updating, using, frequency)

5.3.3. Logical File Structure (record formats, file organization, access methods, rationale, examples)

5.3.4. Physical File Structure (storage device, blocking, organization, access, etc.)

5.3.5. Database Management Subsystems Used (internal or external)

5.3.6. Database Creation and Update Procedure (if NOT by system)

#### 5.4. System Internal Data Structure Specification

5.4.1. Identification of Data Structures

5.4.2. Modules Accessing Structures (creating, updating, using)

5.4.3. Logical Structure of Data (format, organization, access, rationale, examples)

## **6. Module Design Specifications (for each module)**

### **6.1. Module Functional specification**

#### 6.1.1. Functions Performed

#### 6.1.2. Module Interface Specifications (input/output arguments/global variables/files)

#### 6.1.3. Module Limitations and Restrictions

### **6.2. Module operational Specification**

#### 6.2.1. Locally Declared Data Specifications (variable dictionary)

#### 6.2.2. Algorithm Specification (flowchart, pseudocode, decision table, etc)

#### 6.2.3. Description of Module Operation

## **7. System Verification**

7.1. Items/Functions to be Tested

7.2. Description of Test Cases

7.3. Justification of Test Cases

7.4. Test Run Procedures and Results

7.5. Discussion of Test Result



## **8. Conclusions**

8.1. Summary

8.2. Problems Encountered and Solved

8.3. Suggestions for Better Approaches to Problem/Project

8.4. Suggestions for Future Extensions to Project