**COMP 3059 – Capstone Project I**

**Software Requirements Analysis and Design Assignment**

This assignment is an overview to gather the software needs with requirements analysis and help to proceed with the design.

The requirements analysis helps to break down functional and non-functional requirements to a basic design view to provide a clear system development process framework. It involves various entities, including business, stakeholders and technology requirements.

The design is the activity following requirements specification and before programming. Software design usually involves problem solving and planning a software solution.

To work on this assignment you could use the references and a sample template given below. The sample template can be customised to suit the nature of your project.

Reference Readings/Example:

<http://www.uacg.bg/filebank/acadstaff/userfiles/publ_bg_397_SDP_activities_and_steps.pdf>

[www.cse.msu.edu/~chengb/RE-491/Papers/SRSExample-webapp.doc](http://www.cse.msu.edu/~chengb/RE-491/Papers/SRSExample-webapp.doc)

<https://nces.ed.gov/pubs2005/tech_suite/part_2.asp>

Reference template:

[www.tricity.wsu.edu/~mckinnon/cpts322/cpts322-srs-v1.doc](http://www.tricity.wsu.edu/~mckinnon/cpts322/cpts322-srs-v1.doc)

# 1.0 Introduction

The purpose of this document is to outline and describe the high level requirements related to the “How To Train Your Dragon Boat” Application. The sections below will describe the systems features and the purpose of them, as well as describing their functionality through a use of diagrams. This requirements associated with the application are not limited to, but will include the functional and non function requirements of the system, and database requirements for the application.

## Purpose

The Main Purpose of the “How To Train Your Dragon Boat” Application is to provide the George Brown College Dragon Boat Team with an application that is intended to either remove the manual tasks the team captain current performs, or allow the tasks to be completed with more efficiency.

## Scope

* **In-Scope Specifications:** The How to Train Your Dragon Boat Application will be a mobile application supportable by both iOS and Android operating systems. It will also be supported on tablet platforms as well. The application will provide an interactable interface for the creation of customization of boat layouts, with the ability to assign users to the layout. Details about the current layout, including the weight distribution of the members, or their seating preferences, will be displayed along with the layout. This requirement replaces the current method of constructing a layout, which is performed manually, by providing a semi-automated system which will increase efficiency. A map interface will be accessible within the application, designed to assist the team with a valid method of tracking the route in which the boat pursued, as well as the performance of the boat during the current run. Details of the boat’s performance include the distance the boat travelled, the speed or velocity of the boat, and the boat layout that was used during in the current run.

The application will also provide a CRUD interface to manage the users of the dragon boat team, or any other user that may or may not be associated with the dragon boat team. Communication can also be established between users through the use of associated email addresses, eliminating the need to communicate through third party sources or face-to-face conversation. Provided within the application will be an SQLite database that will store all of the users, routes, boat layouts, and the associated information related to each item stored. This will provide a central storage that can be easily accessed by the application and its users.

* **Out-Of-Scope Requirements:** Support for the application will not extend beyond the mobile scope, and will not be supportable by any operating system beyond Android or iOS systems. While the application will use GPS and location for the functionality of the map interface, the application will not use any internet connectivity for functions of the application. The map provided with the application will be used for location tracking only, and will not include any form of custom paths for boat routes. In addition, the map will not include any form of location finding. To be specific, this application will not allow users to use the GPS to find a specific location within an area.

\*(Highlighted statements are new additions to the In Scope/Out or Scope Requirements)

# System Overview

## Project Perspective

The How To Train Your Dragon Boat application is a new self-contained system. Since the George Brown College Dragon Boat team does not have a dedicated system for managing team members and boat layouts, or storing information about users or boat routes along with details, this application will act as a centralized application for all members of the dragon boat team.

## System Context

The System Context describes the resulting software within the business case, including strategic issues in which the system is involved or which it specifically addresses.

## General Constraints

General Constraints identify any business or system constraints that will impact the manner in which the software is to be:

* specified
* designed
* implemented, or
* tested.

## Assumptions and Dependencies

List any assumptions that have been made during the initiation of the project. In addition, list any dependencies that may impact its success or the desired result.

Dependencies:

* Npm install packages
* User must have location enabled

## 3.0 Functional Requirements

This section describes specific features of the software project. If desired, some requirements may be specified in the use-case format and listed in the Use Cases Section.

### 3.1 <Functional Requirement or Feature #1>

* Introduction
* Inputs
* Processing
* Outputs

...

## 3.2 Use Cases

### 3.2.1 Use Case #1 ...

**3.3 Data Modelling and Analysis**

* Normalized Data Model Diagram
* Activity Diagrams
* Sequence Diagrams
* UML Class Diagram

**3.4 Process Modelling**

* Data Flow Diagram

## 4.0 Non-Functional Requirements

The non-functional requirements for a system are typically constraints on the functional requirements – that is, not what the system does, but how it does it (e.g. how quickly, how efficiently, how easily from the user’s perspective, etc.).

### Non-functional requirements may exist for any of the following attributes – Performance, Reliability, Availability, Security, Maintainability, Portability.

Often these requirements must be achieved at a system-wide level rather than at a unit level. State the requirements in the following sections in measurable terms (e.g., 95% of transaction shall be processed in less than a second, system downtime may not exceed 1 minute per day, etc).

## 5.0 Logical Database Requirements

Will a database be used? If so, what logical requirements exist for data formats, storage capabilities, data retention, data integrity, etc?

## 

## 6.0 Other Requirements

Additional requirements, if any.

**7.0 Approval**

The signatures below indicate their approval of the contents of this document.

|  |  |  |  |
| --- | --- | --- | --- |
| Project Role | Name | Signature | Date |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |