

Software System Requirements

Drake Relays Companion App

***William Bartolomucci, Noah Dougen, Jordan Klein,
Nicholas Shipley***

Drake University CS Capstone
Team I

Table of Contents

1. INTRODUCTION	2
1.1 Purpose	2
1.2 Scope	2
1.3 Glossary	3
1.4 Non-Functional Requirements	3
1.4.1 Implementation	
1.5 Use Cases	6
1.5.1 User Interface	7
2. OVERALL DESCRIPTION	8
2.1 Product Perspective	8
2.2 Product Functions	8
2.3 User Characteristics	9
2.4 Constraints and Assumptions	10

1. INTRODUCTION

The introduction to the Drake Relays Companion app software requirements document will serve the purpose of outlining the scope, purpose, and direction of the application. Furthermore, the introduction to this document will give a brief overview of what the application provides to users as well as the owners.

1.1 Purpose

The purpose of this document is to provide an overview of the software system requirements of the Drake Relays Companion app (DRCA). The layout of this document is intended to provide a detailed and concise description of the DRCA and its requirements. Each section is labeled according to what details are intended to be conveyed. The sections iterate in a direction to show the full developmental cycle of the DRCA. In addition to communicating the development life-cycle of the DRCA, there are outlines on details such as functional requirements, external applications that the DRCA will interact with (namely, APIs), and the intended, or beneficial use-cases of the DRCA.

1.2 Scope

The Drake Relays Companion app is a software system that aims to achieve the goal of aiding spectators leading up to, and during, the upcoming Drake Relays. The DRCA will be a helpful tool to interested parties by maintaining a clean and subtle, yet useful interface. The DRCA intends to allow users to track the events of the Drake Relays in real-time as a mobile application. Various features—outlined in subsequent sections—are planned with the user experience in mind.

Event information will include the time the event is taking place, where the event is taking place, and the public's interest in the event. A map will be utilized in the DRCA to disclose to users where on Drake's campus the event is currently, or where it will be, held; a calendar will be present in one of the application's tabs to

ensure that users know the date and time of an event; and the DRCA will include a feature to let users “like” an event—allowing the application to calculate relative interest in an event, and portray the interest to other users.

These features will be connected by a central “hub” of the DRCA. The central hub is intended to be the landing page of the application and will include vital information regarding upcoming Drake Relays events, such as listing the events that are taking place that day and the results of prior events.

The DRCA will provide a better experience for spectators of the Drake Relays. Currently, in order to find events that may be of interest during the Drake Relays, a spectator would be required to navigate multiple web pages on various portions of Drake University’s website; the goal of the DRCA is to streamline the process of finding information about the Drake Relays and provide a more user-friendly experience.

1.3 Glossary

Term	Definition
AWS	AWS stands for Amazon Web Services. AWS will be the database back-end of the Drake Relays Companion app.
DRCA	DRCA stands for Drake Relays Companion Application.
Kivy	Kivy is an open-source Python library capable of creating mobile applications.
RDS	RDS stands for relational database service. In terms of AWS, this service will allow the DRCA to utilize SQL for database querying.
ARDS	ARDS is Amazon Web Service’s relational database feature.

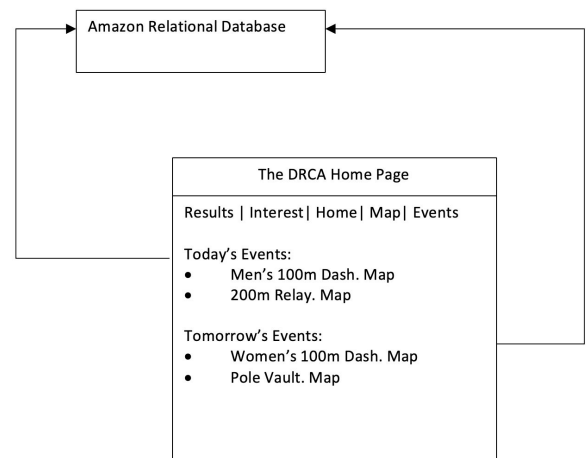
1.4 Non-Functional Requirements

We have chosen to work within the Anaconda network. Files will be created and shared as Jupyter Notebooks. The Jupyter IDE was chosen because it offers an easy way to share code from team member to team member. Another advantage of Jupyter is its split cell functionality. One document can have different “cells” of code that can be run separately. This will make customization of the app easier as different variations of the code can be tested in quick succession.

The app will be developed using the Python language. Specifically, the app will be created in the Kivy Python Framework. This framework was chosen because of the benefits it offers to our project. One of these benefits is its one-size-fits-all app design. A program designed in kivy will work on different platforms such as Android and iOS. Another benefit is that the code itself doesn’t stray far from Python code. This should make it easier to learn in the time frame given for the project.

For the database, we will use Amazon’s AWS. This is a collection of free (for a limited time) services offered by Amazon, including multiple databases. The specific database which will be using is the Amazon Relational Database Service (ARDS). One of the benefits of this database is its use of MySQL. Several members of our team have used or are currently using databases which use MySQL. This familiarity should make implementing this service into our project easier. Another benefit of this database is automated patching and backup.

Fig. 1: ARDS Connection to the DRCA



To populate the database, the information will be created in an Microsoft Excel file. The file will then be read by MySQL code and subsequently placed inside the database appropriately. This is how the bulk of the event data will be entered into the database. For extra additions, such as updates and cancellations, the current owner of the app will have limited database administrator permissions that allow them to enter in and change data as necessary. Ideally, this would be given to the event manager of the Drake Relays. Full database administrator privileges would be given to a member of the Drake IT team.

1.4.1 Implementation

As listed above, the non-functional requirements include the Jupyter IDE, the Python language and Kivy Python Framework, the Amazon AWS ARDS, and some additional requirements such as Microsoft Excel. Each of these, while chosen for their potential benefit to the project, also pose some challenges to the project.

The most obvious challenge is the lack of experience with Kivy. None of the team members have had any prior experience with the Kivy Python Framework. This will inevitably lead to a period where we must learn about the specific quirks of the framework. Luckily, numerous tutorials exist that explain how to use Kivy with step-by-step instructions. Following these tutorials and working in the framework, we should be able to pick up Kivy pretty quickly.

Another challenge that could be faced is user error. This applies mainly to the event manager who would be given control of the app and allowed to make updates. Potentially, someone with this control could accidentally delete important data or upload data in the wrong area, causing an error. Two methods can be used to prevent this. The first is to only give the event manager limited permissions. Full permissions would be given to someone in IT who would be more knowledgeable of database structure. The second method is creating

frequent backups. Considering the size of Relays, backing up all the data shouldn't be hard to manage.

Ideally, the app will function as follows: The event manager will create an excel document containing information for all the Relay events. Once given permission by the Database administrator in IT, the event manager will upload all the events to the app. Students who have downloaded the app can then view the information in different forms (a map of events, a calendar of events, a list of scores, etc.). The only interaction they would have with the data itself is voting "yes" on whether or not they're interested. This would alter an interest column for each event.

When an event was completed, the event manager would update the corresponding scores table with the results. If the event was moved, they would change the location column. If an event was cancelled, they could delete the event. If an event was changed to a different time, they could change the time column. This would continue until Relays eventually ended.

1.5 Use Cases

The Results Feature -- This feature will allow users to keep up-to-date with how their respective universities are faring in the Drake Relays.

The Map Feature -- This feature will allow users to quickly take a glance at where events are currently being held, or will be held.

The Interest Feature -- This feature will allow users to see which events might harbor the best atmosphere when attending the event.

The Events Feature -- This feature will allow users to keep track of events, and ensure that they will not miss events that they were planning attending.

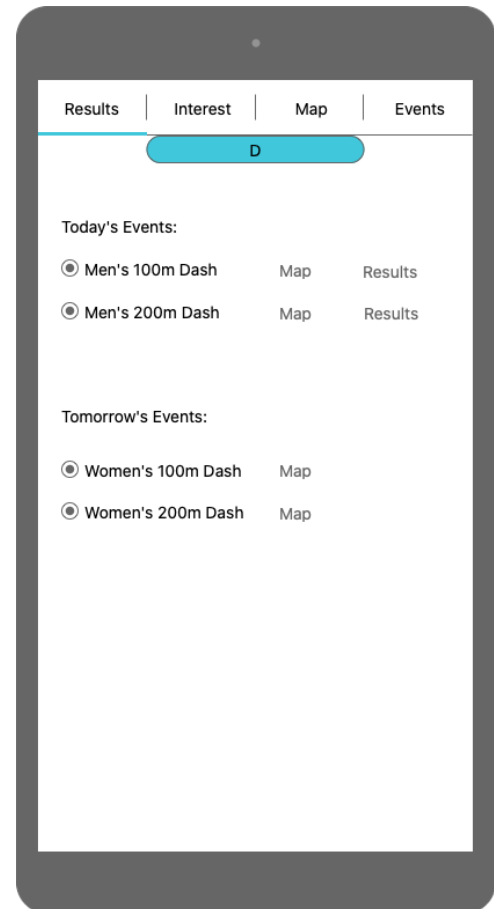
Fig. 2: Rudimentary User Interface; Landing Page

1.5.1 User Interface

The following image is a rudimentary mark-up of the design of the mobile application's landing page. Along the top, there are tabs which, when clicked, take the user to a different part of the application. These include a results page that show the results of previous events, an interest page that shows which events are accumulating a lot of interest, a map, which will show a map of campus with pins that show where events are currently taking place, and an events page, which overviews the current and upcoming events.

The landing page, as shown in the figure, shows the events that are upcoming. To the left hand side of the upcoming events is a radio button. When this button is pressed, it indicates that the user is interested in going to the event. To the right hand side of the listed events are links to the map and results (if they are available). The map link will redirect the user to a campus map, and indicate, via a pin, where on campus the event is being held.

When the “interest” button is pressed, there will be a query sent to the database linked to the DRCA indicating that someone is interested. This portion of the database will be used to populate the “interest” page of the application. Depending on the amount of anticipated users, there will be a calculus as to what constitutes an event with a high amount of interest; whatever that may be, the events which garner a lot of interest will be highlighted in the aforementioned page.



As for the results page in the application, the event manager will enter various information into a table in the database. The information that could be included are race times, who in the race made it to the next heat, if a record was broken, and other factors that may be helpful for users to know. The results page will then connect with ARDS and output the results in an easy-to-read manner for the user.

2. OVERALL DESCRIPTION

This section serves to give an overview of the entire system of the Drake Relays Companion app. The functions of this app will be thoroughly explained as well as details regarding the multiple systems that interact with this app. This section also includes a description of the different functions that each type of stakeholder will have access to within these systems. An examination of the constraints and assumptions will also be provided.

2.1 Product Perspective

The DRCA will retrieve results and event information through the database hosted on AWS. The database can be edited by the event manager, and the changes will be shown to the user of the application real-time. The DRCA is further connected to ARDS with the “interest” feature.

Results and event information will flow from the database to the application, while the “interest” feature will flow from the application to the database.

2.2 Product Functions

The functions of this app include the following:

- A main hub that lists events which will take place that day.

- A calendar which lists every event by date and time.
- A map of campus that shows where events are currently occurring.
- An interest meter for each event (a student can “star”, or “like” an event, showing how many people are interested in it).
- Gauge the current business of the Drake relays by utilizing Google’s traffic API.
- A main hub that lists results of events.
- The ability to upload events by date and location.
- The ability to highlight certain events, like those that will take place soon.
- The ability to edit or cancel events on the fly.
- The ability to edit results of events.
- A login system to keep users confined within certain limitations.

2.3 User Characteristics

There are three intended users of the DRCA which include administrators, event management, and general users of this application. Each intended user has a specific purpose in mind when interacting with the DRCA and thus, different users have different abilities within the application.

Administrator users serve the purpose of ensuring accurate information is presented across the application. They have access to all systems within the application and access to both event management and general user functions.

The primary focus of the event management user is to allow changes to be made to the events themselves. Unique login capabilities within the app will allow event managers to carry out these changes. Specific changes include updating the details of events, creating or cancelling events, and entering/managing the results of competitions in real time as events are completed.

General users of the application are very limited to specific functions within the application. These users will be able to log in for full access to general user functions or continue as guests with limited access. The use of the mobile app serves to help the user find events, results, and locations and overall enjoy the events that take place at the Drake Relays. General users have no ability to make any alterations within the DRCA.

2.4 Constraints and Assumptions

The DRCA is limited by a user's connection to the internet. Multiple functions will be unavailable without internet connection or data may not be updated.

Allowing users to access and request information from the database within the app can create slow response times with multiple users requesting information at the same time.

This application is made with the idea that users will access the application via a smartphone device. Other devices or older models may present software issues with the application and cause limitations.