

# Harvest Requirements and Design Documentation

May 10, 2018

**Team Name:** Binary Ninjaz

**Team Members:**

Teboho Mokoena (u14415888)

Sizo Duma (u15245579)

Letanyan Delon Arumugam (u14228123)

John Ojo (u15096794)

Kevin Reid (u15008739)

Shaun Yates (u16007493)

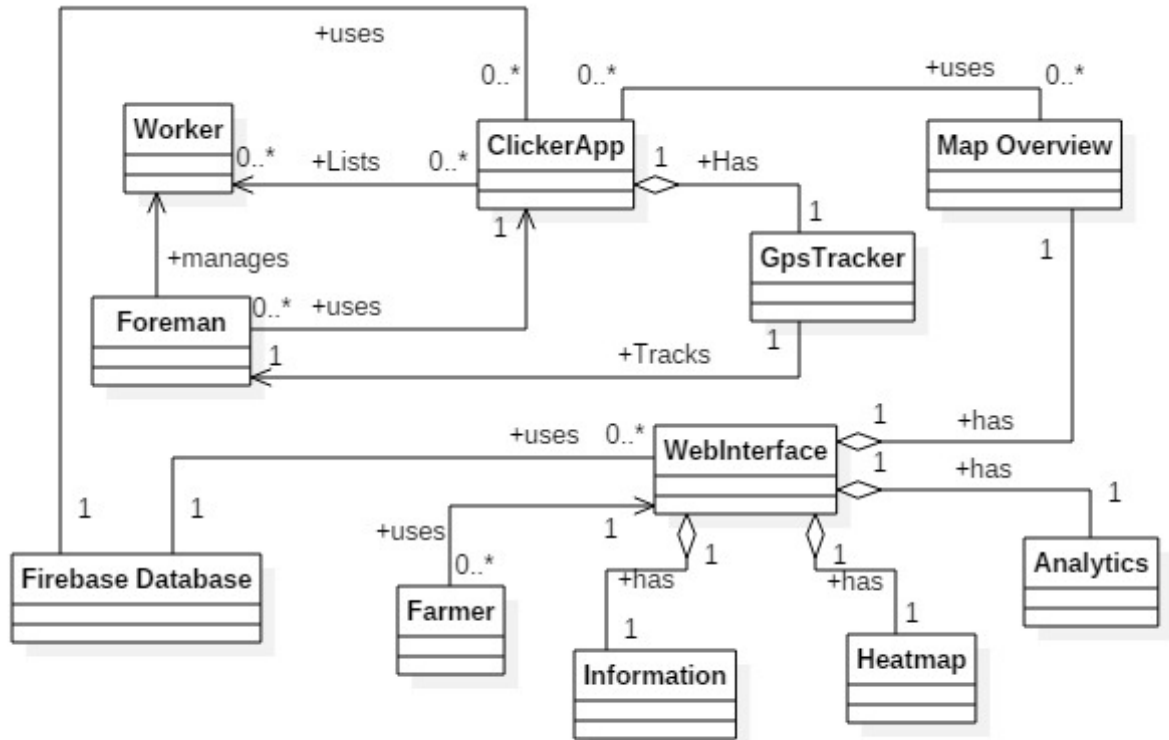
## **Introduction**

The goal of the project is to design and develop a software system to be used by the farming community in the facilitation and management farming. The software system in development (Harvest) will be consisting of: an Android application, iOS application, and a website.

### **Functional Requirements:**

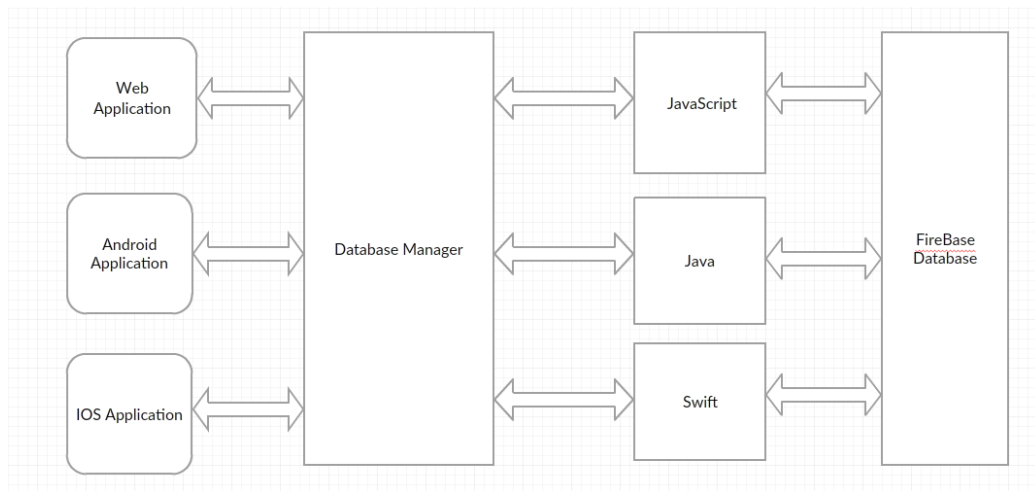
- On sign in Harvest shall verify credentials against the database before signing them in.
- Harvest shall create a profile on the database for the user through a registration form.
- Harvest shall measures yield, by combining a clocking system with yield data, through the input view of the mobile interface.
- The Harvest mobile interface shall make use of GPS data and based on weight and location assumptions, it must give approximate yield estimates not only for each orchard, but for each approximate location where the data was entered.
- The Harvest mobile interface shall make use of analytics to measure worker performance.
- The Harvest website shall display data on the farm on a heat map.
- The Web interface shall show detailed information about produce. (e.g. cultivar, year planted)
- The Harvest system shall track the foreman through their phone/tablets GPS.
- The Harvest system shall do administrative tasks through the web interface.
- The Harvest website shall give the farmer/administrator a real time map view of the foreman's location.

## The Domain Model

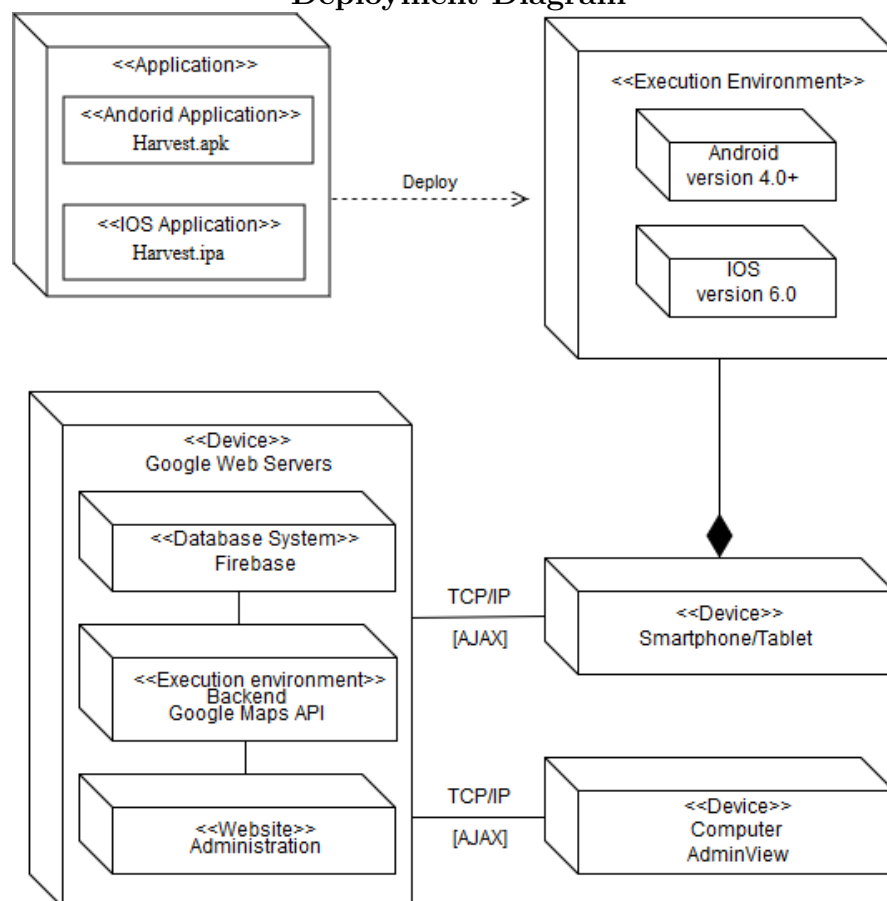


## Architectural Design Process

The architectural structural design of our the Harvest system is a Persistence Framework. The system has 3 main subsystems (Android Application, IOS Application and the website) all heavily reliant on the database system providing them with efficient object storage and retrieval without the need for implementation detail.



### Deployment Diagram



### 0.0.1 Determining Design Objectives

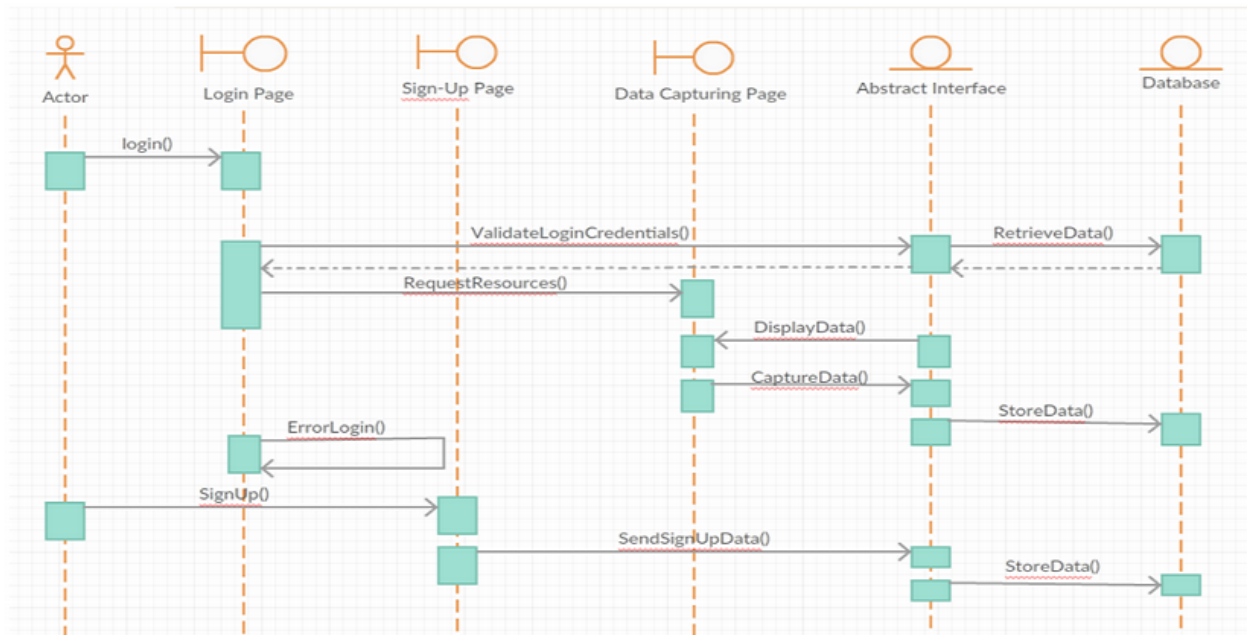
- The design is aimed to provide accessibility, data integrity and optimize performance.
- The design is also aimed at producing a user friendly easy to use system.
- The design is aimed at easing facilitation and automation of work.

### 0.0.2 Determining Type of System

The system under development is identified as an Object persistence subsystem (also known as a Database subsystem), because heavily focuses on efficient information storage and retrieval . It also hides the database details such as implementation from the subsystems, and therefore uses an abstract interface to manipulate data.

### 0.0.3 Specifying Subsystem Functions, Interfaces, and Interaction Behaviour

- **The Application Interface:** This interface will have the functionality for login in and signing up of foremen. It will have yield data recording functionality and worker performance capturing functionality.
- **The Web Application Interface:** The administrative tasks will be carried out on the Interface (Like adding worker and Orchard Details per farm), as well as tracking of workers and performance.
- **The Database Interface:** This interface will maintain strict access, it will only be accessed through an abstract interface (Backend) only to store and retrieve data.



#### 0.0.4 Reviewing the architectural design

- The Harvest application is accessible from a smartphone/tablet/desktop computer.
- The Harvest system checks whether the user is registered before signing them in.
- Harvest creates a new profile of the user when registering them.
- Harvest does measures yield, by combining a clocking system with yield data, through the input view of the mobile interface.
- Harvest does facilitate GPS data filtering and Web Functionality
- The software application does facilitate the idea of tracking

#### 0.0.5 Technology Used

- Database System : Firebase
- IDE's: Xcode, Android studio
- Programming languages: XML (Android UI); Java (Android Backend); Swift (IOS); HTML, CSS, JavaScript (Website)