System and Software Architecture Description (SSAD)

Farmworkers Safety App

Team 09

TEAM MEMBER NAME	ROLES
Shobhit Agarwal	Project Manager Life Cycle Planner System Architect
Akshay Aggarwal	System Architect Prototype Developer Feasibility Evidence Analyst
Viraj Sahai	Prototype Developer Feasibility Evidence Analyst
Vahagen Sinanian	Operational Concept Developer NDI Analysis Personas
Juan Andrade	Requirements Engineer Prototype Developer Operational Concept Developer
Basir Navab	Life Cycle Planner Project Manager
Marko Djuliarso	Independent Verification and Validation Quality Focal Point

10/17/2016

VERSION HISTORY

Date	Author	Version	Changes Made	Rationale
10/12/2016	Shobhit Agarwal	1.0	Sections 1 and 2 diagrams added	Draft for the FCR ARB Submission
10/16/2016	Viraj Sahai	1.1	Sections 1 and 2 completed	Final copy for FCR ARB Submission
10/17/2016	Shobhit Agarwal	1.2	Adding section 2.2, adding header and footers, and formatting the document	Final copy for FCR ARB submission

Table of Contents

Table of Tables

Table of Figures

1. Introduction

1.1 Purpose of the SSAD

The purpose of this SSAD is to document the design of the Farmworker Safety App. The SSAD will be a guiding tool for analysis and detailed description of the system architecture. It would serve as a reference for the future software developer/maintainer to understand and enhance the design in the future.

1.2 Status of the SSAD

-Sections 1 and 2 have been completed for Foundations Commitment Package.
-All other sections will be completed at a later date.
-Naming conventions are likely to change in further revisions (refer to section 2.2 System Analysis Rationale for further details).

2. System Analysis

2.1 System Analysis Overview

The main purpose of the Farm Worker App is to provide notifications to farmworkers and farmers about the temperature levels if it exceeds a threshold so that they can take breaks and save themselves from heat related stress. The app will also collect feedback from the farmers regarding their health conditions so that they can be monitored for symptoms of heat related stress. It will also include emergency services like one touch call to CalOsha, 911 and other emergency contacts. It may also provide a platform for contractors and farmers to search for available farmworkers looking for jobs.

2.1.1 System Context

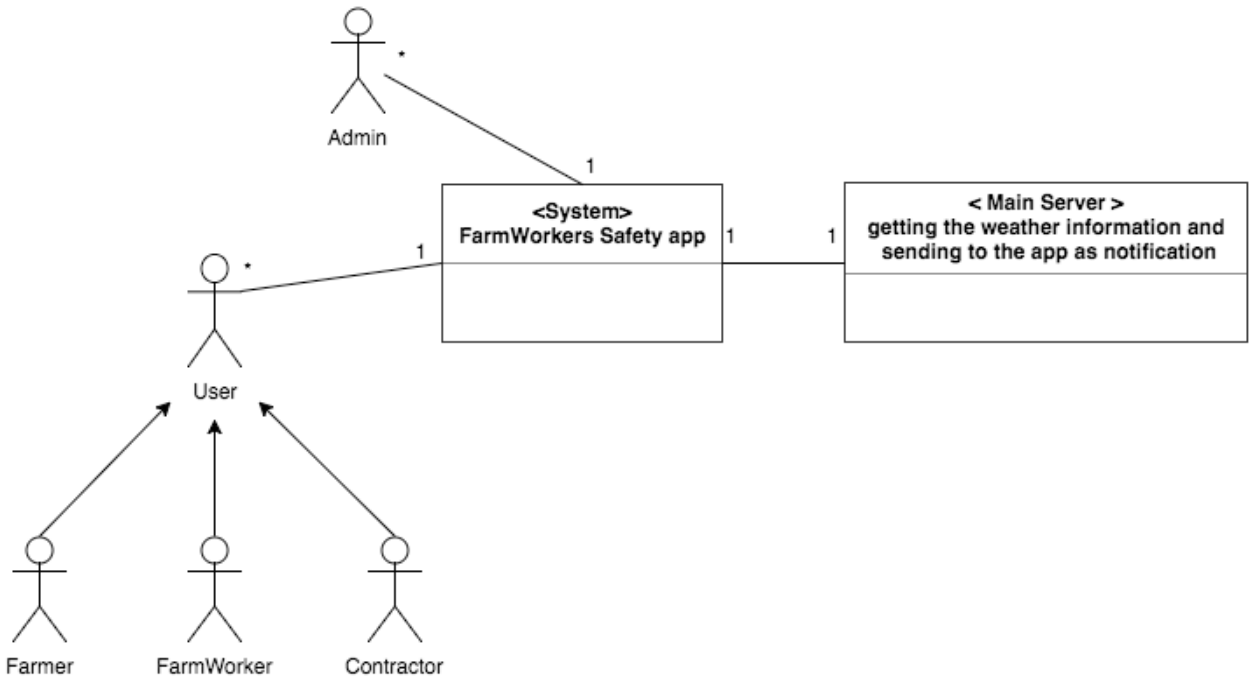

Figure 1: System Context Diagram

Table 1: Actors Summary

Actor	Description	Responsibilities
Admin	Administrators to manage the system and functionalities	Control user access rights; manage data; post content; manage users
User (Farmer)	Users that own farm	Manage farmworkers; register new farmworkers to the farm; hire farmworkers
User (Contractor)	Users that manage farmworkers	Manage farmworkers; register new farmworkers to the farm; hire farmworkers
User (Farm Worker)	Users that work on the farms	Submit feedback; take quizzes; update location; get himself educated; take preventive measures

2.1.2 Artifacts & Information

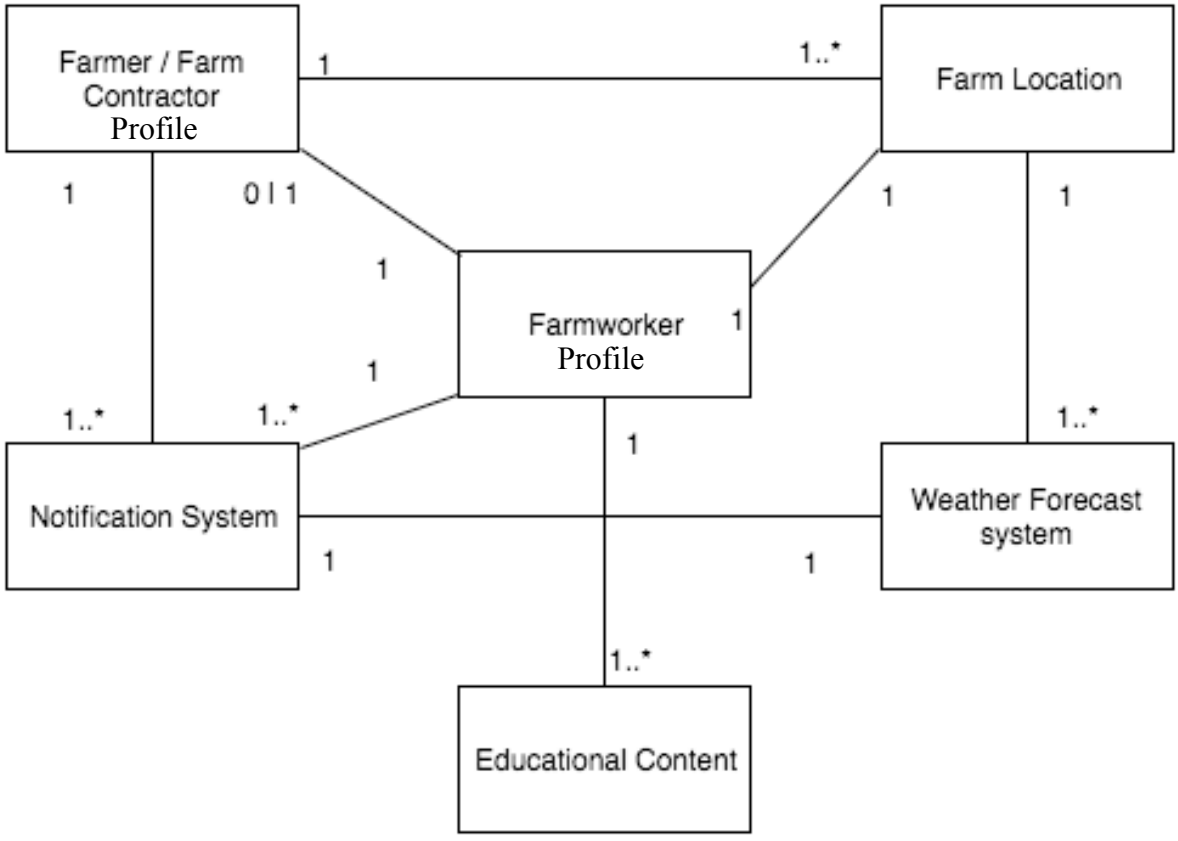

Figure 2: Artifacts and Information Diagram

Table 2: Artifacts and Information Summary

Artifact	Purpose
Farmworker Profile	Contains information regarding the farmworkers
Farmer Profile	Contains information regarding the farmer/contractor
Farm Location	Location of the farm at which the farmworker is working
Notification System	Provides notifications to the farmworkers regarding temperature, quizzes and feedback
Educational Content	Videos and Quizzes to educate the farmworkers regarding heat related stress and prevention practices
Weather Forecast System	Fetches weather information to be able to send notifications to the farmworkers

2.1.3 Behavior

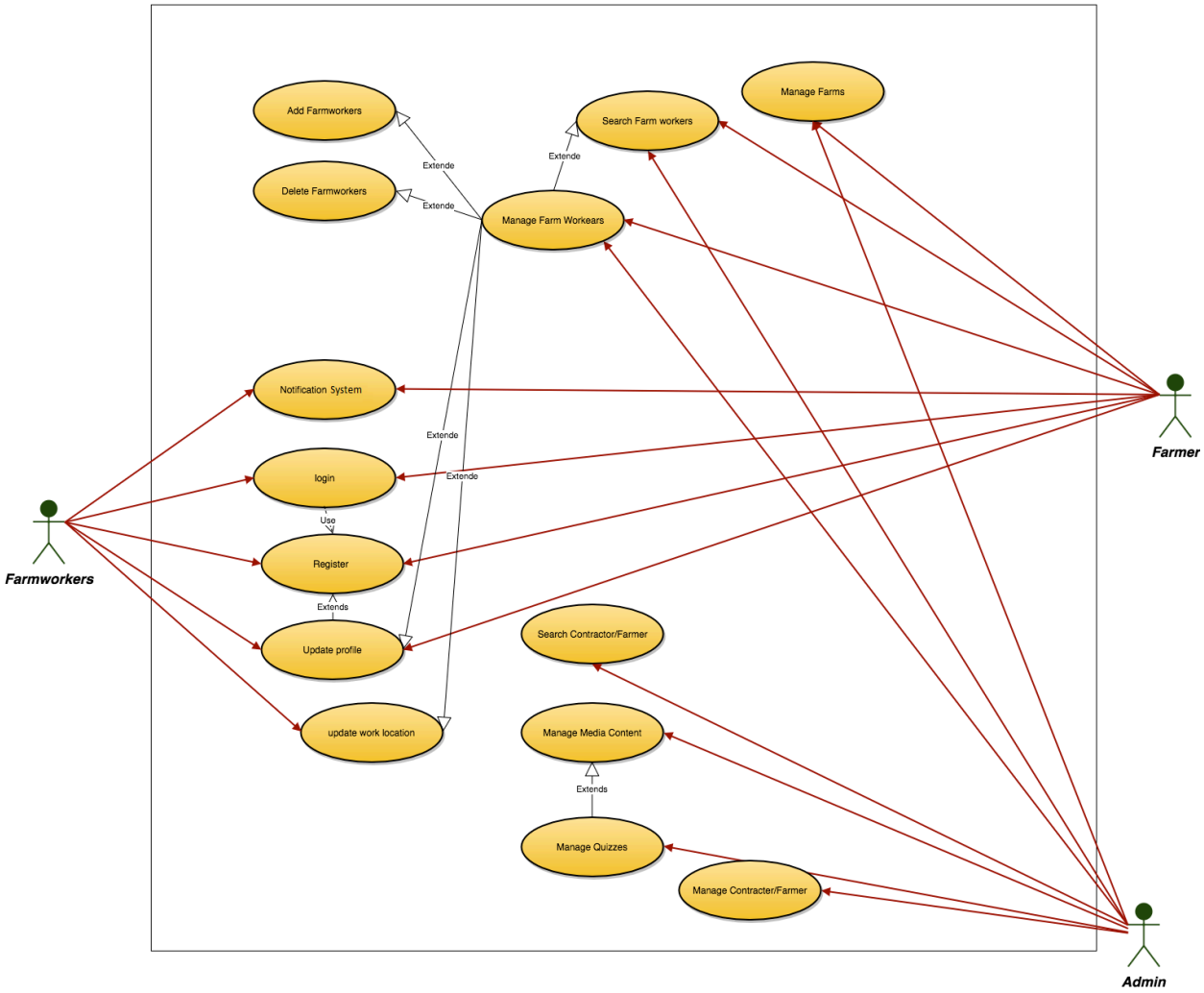

Figure 3: Process Diagram

2.1.3.1 Authentication

2.1.3.1.1 Login

Table 3: Process Description-login

Identifier	login
Purpose	Determine if a person logging in to the system can be authenticated, and, if so, what the person's privileges are as a user.
Requirements	Authorization and Authentication
Development Risks	None
Pre-conditions	System database is properly initialized. Login functionality is successfully implemented.
Post-conditions	If user exists in the system the user shall be logged in with appropriate access rights

Table 4: Typical Course of Action-login: successful

Seq#	Actor's Action	System's Response
1	User enter username and password and click enter	
2		Check whether username and password is true.
3		Redirect user to user's main page

Table 5: Alternate Course of Action-login: fail

Seq#	Actor's Action	System's Response
1	User enter username and password and click enter	
2		Check whether username and password is true.
3		Return error message like "user name or password incorrect" or "user doesn't exist"

2.1.3.1.1 Logout

Table 6: Process Description-logout

Identifier	logout
Purpose	Logout the user from the system
Requirements	Authorization and Authentication
Development Risks	None
Pre-conditions	User is a valid system user User is logged in to the system and the session exists.
Post-conditions	The login session is terminated.

Table 7: Typical Course of Action-logout: successful

Seq#	Actor's Action	System's Response
1	User clicks on the logout button	
2		User is logged out of the system and a message is displayed saying " You have been successfully logged"

Table 8: Alternate Course of Action-logout: fail

Seq#	Actor's Action	System's Response
1	User logs out an expired session	
2		Return session expired message and return to log in page

2.1.3.2 Registration

2.1.3.2.1 User Registration

Table 9: Process Description-Registration

Identifier	Registration
Purpose	Register user to the system
Requirements	Internet connection on user end. Registration webpage.
Development Risks	None
Pre-conditions	User is not an already registered user

	User has loaded the registration page
Post-conditions	User is registered to the system

Table 10: Typical Course of Action-Registration: successful

Seq#	Actor's Action	System's Response
1	User fills in the registration form	
2		User is registered into the database
3		Redirect user to the main profile page

Table 11: Alternate Course of Action-Registration: fail

Seq#	Actor's Action	System's Response
1	Already a registered user tries re-registering	
2		Return a "user already registered" error

2.1.3.3 Profile Management

2.1.3.3.1 Update Profile

Table 12: Process Description-Update Profile

Identifier	Update profile
Purpose	Update personal user profile
Requirements	User should be registered
Development Risks	None
Pre-conditions	User should be logged in
Post-conditions	Profile data is updated in the database

Table 13: Typical Course of Action-Update Profile: successful

Seq#	Actor's Action	System's Response
1	User updates the required field	
2		User data is updated in the database and a message is displayed as "Profile Updated"

Table 14: Alternate Course of Action-Update Profile: fail

Seq#	Actor's Action	System's Response

1	User updates unique data like email to one that is already registered	
2		Return a "Update failed. Email already registered to another user" error

2.1.3.3.2 Update Location

Table 15: Process Description-Update Location

Identifier	Update work location
Purpose	Update location of user so as to be able to send notifications
Requirements	User should be registered
Development Risks	None
Pre-conditions	User should be logged in
Post-conditions	Location data is updated in the database

Table 16: Typical Course of Action-Update Location: successful

Seq#	Actor's Action	System's Response
1	User updates the location using place name or zip-code	
2		Location data is updated in the database and a message is displayed as "Location Updated"

Table 17: Alternate Course of Action-Update Location: fail

Seq#	Actor's Action	System's Response
1	User input invalid place or zip-code	
2		Return a "Invalid Location input" error

2.1.3.4 Farm Management

2.1.3.4.1 Manage Farms

Table 18: Process Description-Manage farms

Identifier	Manage Farms
Purpose	Add/Delete Farms; Update Farm attributes like area, facilities etc.; View farm details and list of employed workers
Requirements	User should be registered as a farmer
Development Risks	None
Pre-conditions	User should be logged in User should be a farmer
Post-conditions	Farm data is updated in the database

Table 19: Typical Course of Action-Manage Farms: successful

Seq#	Actor's Action	System's Response
1	User performs the required changes	
2		Farm data is updated in the database and a message is displayed as "Farm data updated"

Table 20: Alternate Course of Action-Manage Farms: fail

Seq#	Actor's Action	System's Response
1	User inputs invalid data like invalid zip-code	
2		Return a "Update failed. Invalid zip-code" error

2.1.3.5 Farmworker Management

2.1.3.5.1 Add Farmworker

Table 21: Process Description-Add Farmworker

Identifier	Add farmworkers
Purpose	Add Farmworkers to the Farmer's farm
Requirements	User should be registered as a farmer
Development Risks	None
Pre-conditions	User should be logged in User should be a farmer User should add farmworkers to valid farm
Post-conditions	Farmworker data is updated in the database

	Corresponding update also shows in the Farmworkers profile

Table 22: Typical Course of Action-Add Farmworker: successful

Seq#	Actor's Action	System's Response
1	User adds farmworkers	
2		Farmworker data is updated in the database and a message is displayed as "Farmworker added"

Table 23: Alternate Course of Action-Add Farmworker: fail

Seq#	Actor's Action	System's Response
1	User adds a farmworker who is not registered to the system	
2		Return a "Farmworker not registered" error

2.1.3.5.2 Delete Farmworker

Table 24: Process Description-Delete Farmworker

Identifier	Delete farmworkers
Purpose	Delete Farmworkers from the Farmer's farm
Requirements	User should be registered as a farmer
Development Risks	None
Pre-conditions	User should be logged in User should be a farmer User should delete farmworkers from valid farm
Post-conditions	Farmworker data is updated in the database Corresponding update also shows in the Farmworkers profile

Table 25: Typical Course of Action-Delete Farmworker: successful

Seq#	Actor's Action	System's Response
1	User deletes farmworkers	
2		Farmworker data is updated in the database and a message is displayed as "Farmworker deleted"

Table 26: Alternate Course of Action-Delete Farmworker: fail

Seq#	Actor's Action	System's Response
1	User deleted a farmworker who is not registered to the system	
2		Return a "Farmworker not registered" error

2.1.3.5.3 Search Farmworker

Table 27: Process Description-Search Farmworker

Identifier	Search farmworkers
Purpose	Search Farmworkers in the Farmer's farm to display their info
Requirements	User should be registered as a farmer
Development Risks	None
Pre-conditions	User should be logged in User should be a farmer
Post-conditions	Farmworker is fetched from the database

Table 28: Typical Course of Action-Search Farmworker: successful

Seq#	Actor's Action	System's Response
1	User searches farmworkers	
2		Farmworkers data is fetched from the database and displayed

Table 29: Alternate Course of Action-Search Farmworker: fail

Seq#	Actor's Action	System's Response
1	User searches for farmworker not registered in the system	
2		Return a "Farmworker not found" error

2.1.3.6 User Management

2.1.3.6.1 Manage Users

Table 30: Process Description-Manage Users

Identifier	Manage Contractors/Farmworkers
Purpose	Add/Delete users; Update user data
Requirements	User should be registered as an admin
Development Risks	None
Pre-conditions	User should be logged in User should be an admin
Post-conditions	Farmworker data is updated in the database Corresponding update also shows in the Farmworkers profile

Table 31: Typical Course of Action-Manage Users: successful

Seq#	Actor's Action	System's Response
1	User updates other users' data	
2		Users' data is updated in the database and a message is displayed as "User Updated"

Table 32: Alternate Course of Action-Manage Users: fail

Seq#	Actor's Action	System's Response
1	User updates unique data fields leading to table constraints violations	
2		Return a "User with same data already present" error

2.1.3.6.2 Search Users

Table 33: Process Description-Search Users

Identifier	Search Contractors/Farmworkers
Purpose	Search users
Requirements	User should be registered as an admin
Development Risks	None

Pre-conditions	User should be logged in User should be an admin
Post-conditions	Farmworker data is fetched from the database

Table 34: Typical Course of Action-Search Users: successful

Seq#	Actor's Action	System's Response
1	User searches for other users' data	
2		Users' data is fetched from the database and displayed

Table 35: Alternate Course of Action-Search Users: fail

Seq#	Actor's Action	System's Response
1	User searches for a non-existent user	
2		Return a "User not found" error

2.1.3.7 Content Management

2.1.3.7.1 Manage Media Content

Table 36: Process Description-Manage Media Content

Identifier	Manage Media Content
Purpose	Add/Delete media content
Requirements	User should be registered as an admin
Development Risks	None
Pre-conditions	User should be logged in User should be an admin
Post-conditions	Media content shows as a post to farmworkers Notifications are sent to farmworkers

Table 37: Typical Course of Action-Manage Media Content: successful

Seq#	Actor's Action	System's Response
1	User adds educational video	
2		Video is added and a "Video Added" message is displayed

Table 38: Alternate Course of Action-Manage Media Content: fail

Seq#	Actor's Action	System's Response
1	User adds content in incorrect format	
2		Return a "Unsupported file format" error

2.1.3.7.2 Manage Quizzes

Table 39: Process Description-Manage Quizzes

Identifier	Manage Quizzes
Purpose	Add/Delete Quizzes
Requirements	User should be registered as an admin
Development Risks	None
Pre-conditions	User should be logged in User should be an admin
Post-conditions	Quiz shows as a post to farmworkers Notifications are sent to farmworkers

Table 40: Typical Course of Action-Manage Quizzes: successful

Seq#	Actor's Action	System's Response
1	User adds quiz	
2		Quiz is added and a "Quiz Added" message is displayed

Table 41: Alternate Course of Action-Manage Content: fail

Seq#	Actor's Action	System's Response
1	User add quiz with missing options	
2		Return a "Incomplete quiz" error

2.1.3.8 Notification System

2.1.3.8.1 Notification System

Table 42: Process Description-Notification System

Identifier	Notification System
Purpose	Send notification to the user
Requirements	User should be registered as a user
Development Risks	None
Pre-conditions	User should be logged in
Post-conditions	Notification is sent to the user and displays in status bar

Table 43: Typical Course of Action-Notification System: successful

Seq#	Actor's Action	System's Response
1		System sends temperature and other notifications
2	Users reacts accordingly	

2.1.4 Modes of Operation

Farmworker app will operate in only one mode, so nothing further need be said of modes of operation.

2.2 System Analysis Rationale

Administrator registration:
Admin registration has not been included explicitly, as this would be created manually for each admin. Number of admins should be limited and should not be shown in the UI. Only existing admins should be able to add new admins.

User type - Farmer and Contractor:
Both these user types have the same functionality and hence are not being shown explicitly. The only difference in terms of the app between these users is that farmers can manage contractors as well, which will not be covered in the near future, and would be handled manually from the backend for the time being.

Admin can send notification:
Admin has the functionality to send manual notification to the users of the app, but this has not been shown in the use case diagram as this would be one of the implicit feature of the admin user.