Feasibility Evidence Description (FED)

Farmworkers Safety Application

Team 9

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

October 17, 2016

Version History

Date	Author	Version	Changes made	Rationale
10/11/16	Akshay Aggarwal	1.0	• Original version of the FED	• Preparation for the FCR ARB
10/17/16	Akshay Aggarwal	1.1	• FED Sections 1-5 Completed	• FC Package Submission

Table of Contents

Table of Tables

Table of Figures

No figures in this report at this time

1. Introduction

1.1 Purpose of the FED Document

The Feasibility Evidence Description document aims to quantify the benefits of and evaluate all aspects of the Farmworkers Safety Application project in an evidence-driven manner.

The feasibility study presented here aims to uncover the strengths and weaknesses of our proposed plan for the Farmworkers project, list opportunities and risks associated with our project, and quantify the resources required to make our project a success.

In our feasibility analysis, we are mainly concerned with establishing the technological, economic, and operational feasibility of our proposed solutions for the Farmworkers safety application.

1.2 Status of the FED Document

This document is a revised draft of the FED document that was prepared for the FCR ARB. It now includes updated information about costs involved in the project, benefits of the project, an analysis of the return on investment from the completion/execution of the project, feasibility of the project, risk assessment, and an analysis for the selection of NDIs used in the project.

2. Business Case Analysis

Assumptions:

Farmers have phones and Internet connectivity; They can use their phones at work.
They have nearby access to water and shade.
Farmers want to improve working conditions.
Contractors and farmers are obligated to provide safety for farmworkers.

Stakeholders (Who?)	Initiatives (What?)	Value Proposition (Why?)	Beneficiaries (For Whom?)
- Developer - Maintainer - Farmworker - Contractors - Farmers - Regulators	- Develop the system - Keep the system up and running - Setup a profile with correct phone number and accurate location - Update locations of farmworkers based on their farm assignments - Manage farmworkers through profiles and provide feedback - Set standards for farmer safety	- Provide temperature based notifications. - Educate farmers and improve their quality of life. - Improve productivity of farmworkers.	- Farmworkers - Consumers of farm products - Farmers and contractors

Cost:	Benefit (Metrics):
- Development cost - Maintenance cost - COTS products: weather, and SMS API - Backend server and database	- Decrease the number of people who suffer from heat illness. - Increase the productivity of the farms.

2.1 Cost Analysis

2.1.1 Personnel Costs

Table 1: Personnel Costs

Activities	Time Spent (in hours)
Valuation and Foundations Phases (CSCI 577a)	
Client/Team Communications via email, phone, BaseCamp, and messaging: 4hr/wk * 12 wks * 2 people	96
Win-Win Negotiation Sessions: 2 sessions * 1 hr/session * 2 people	4
Meeting with Farmer: 1.5 hr * 1 person	1.5
Meetings with Farmers and Contractors: 8 hrs * 5 people	40
Architecture Review Boards: 1.5 hrs * 2 sessions * 2 people	6
Development and Operations Phases (CSCI 577b)	
Client/Team Communications via email, phone, BaseCamp, and messaging: 4hr/wk * 12 wks * 2 people	96
Architecture Review Boards: 1.5 hrs * 2 sessions * 2 people	6
Deployment of system: 5 hrs * 3 people	15

Training of system admins and stakeholders: 3 hrs * 4 people	12
Total	276.5

2.1.2 Hardware and Software Costs

Table 2: Hardware and Software Costs

Type	Cost	Rationale
Ownership Cost	Cost of COTS + Web server	Need to deliver weather and SMS services and store data
Maintenance Cost	$0	No foreseeable maintenance costs
Hardware	$0	No foreseeable hardware costs
Total	Cost of COTS + Web Server	The system does not have a fixed cost. Costs are variable depending on the number of deployed farms.

2.2 Benefit Analysis

The main benefits of the Farmworkers Safety Application/System are the following:

- Increase transparency of working conditions to prevent heat-related injuries and illnesses
- Improve access to educational material that details best practices for avoiding heat-illness and that describes how to be more healthy

Table 3: Benefits of the Farmworkers Safety System

Current activities & resources used	% Reduce	Time Saved (Hours/Year)
There are currently no standard activities, procedures, or resources used to monitor temperature.		
Total		

2.3 ROI Analysis

As there are no known financial or other measures for efforts being made to monitor working conditions or to educate farmworkers, it is not possible to calculate an ROI for our project at this time. Our team will be visiting a farm on October 29th, and we will have the opportunity to acquire information regarding the number of man-hours and resources spent in monitoring the health of farmworkers and educating farmworkers at that time.

Table 4: ROI Analysis

Year	Cost	Benefit (Effort Saved)	Cumulative Cost	Cumulative Benefit	ROI

3. Architecture Feasibility

3.1 Level of Service Feasibility

Table 5: Level of Service Feasibility

Level of Service Requirement	Product Satisfaction
LOS-1: The system shall be scalable to up to at least the 400,000 farmworkers in California	Product Strategies: Paid COTS are capable of handling large volumes of concurrent users
	Process Strategies: Monitor number of concurrent users and purchase premium COTS plans as required
	Analysis: Monitoring of users and purchase of COTS will allow us to proactively scale-up our system
LOS-2: The system shall have cross platform and cross system capabilities	Product Strategies: Use SMS and a Web Application for platform-independent support
	Process Strategies: Deploy new features of web application before porting for mobile applications.
	Analysis: Using a web-first strategy and deploying on multiple platforms will enable all types of users to access the service
LOS-3: The system shall not be down for more than 24 hours in a month	Product Strategies: Choose COTS and service providers that offer priority support and that have a proven record of quality
	Process Strategies: Use Sundays for preventative maintenance and deployment of new versions of application. Beta test new versions of application before public releases.
	Analysis: Performing preventative maintenance is a proven method of avoiding downtime, and selecting well-reputed COTS ensures critical components of our system remain online

3.2 Capability Feasibility

Table 6: Capability Requirements and Their Feasibility Evidence

Capability Requirement	Product Satisfaction
CR-1: Fetch weather	Software/Technology used: Weather API (darksky.net)
	Feasibility Evidence: Prototype Weather API integration
	Referred use case diagram: Use Case Diagram 2.1.5 in the SSAD document
CR-2: Send text based notifications	Software/Technology used: SMS API (Nexmo)
	Feasibility Evidence: Prototype SMS API integration
	Referred use case diagram: Use case diagram 2.1.5 in the SSAD
CR-3: Host Educational Media Content and Store User Profiles	Software/Technology used: Database
	Feasibility Evidence: Evaluate different database providers (Amazon, Google)
	Referred use case diagram: Use case diagram 2.1.5 in the SSAD

3.2 Evolutionary Feasibility

Table 7: Evolutionary Requirements and Their Feasibility Evidence

Evolutionary Requirement	Product Satisfaction
ER-1: Enable dynamic educational content	Software/Technology used: Content management system
	Feasibility Evidence: Create a a prototype that enables system administrators to upload new content and to specify where this content should be displayed

4. Process Feasibility

Table 8: Rationales for Selecting Architected Agile Model

Criteria	Importance	Project Status	Rationales
30 % of NDI/NCS features	3	3	We will use NDI/NCS features to access weather information for farms across California and to send text message notifications to farmworkers
Single NDI/NCS	1	1	We will certainly be using more than one NDI/NCS
Unique/ inflexible business process	1	1	The business processes are very flexible
Need control over upgrade / maintenance	2	2	The system should be able to be updated and upgraded in order to support more and more farmworkers
Rapid deployment	0	0	We will first develop and deploy a system on one small test farm before expanding the solution to cover farms in Southern California then in California as a whole.
Critical on compatibility	0	0	There are no compatibility issues between the selected NDIs/NCIs or with other technologies used in the project
Internet connection independence	1	1	The system will not be independent of an Internet connection. An Internet connection is required to fetch weather information and to send automated text messages to users of our system
Need high level of services / performance	3	3	It is critical that we maintain a high level of service as our system is used for ensuring the safety of thousands of laborers.
Need high security	2	2	Controlling access to our

			system is not of high priority, although maintaining the privacy of our users is crucial.
Asynchronous communication	2	2	Asynchronous communication is key to providing our service at scale. If calls for fetching weather information happen sequentially instead of in parallel, our system may suffer from high latency
Be accessed from anywhere	3	3	It is essential that our system enable high-temperature alerts for any region within California, later within the United States, and, at some point, potentially the world
Critical on mass schedule constraints	0	0	There are no schedule constraints as of yet. However, the system must be deployed as soon as possible.
Lack of personnel capability	0	0	The groups who will work on this project will be proficient in many languages and confident with a broad-range of topics in Computer Science and Software Engineering
Require little upfront costs	2	2	Our project has a limited but non-zero budget. We are authorized to select COTS that have some costs associated with them, provided that such spending is justified
Require low total cost of ownership	2	2	There is a mild cost of ownership caused by the pay-per-use of NCIs used in the project and for web-hosting
Not-so-powerful local machines	3	3	Machines used in practice will be assumed to have limited computational power, and, as such, most processing needs to take place at the server-side

5. Risk Assessment

Table 9: Risk Assessment

Risks	Risk Exposure			Risk Mitigations
	Potential Magnitude	Probability Loss	Risk Exposure	
Design Constraints: The app will cater to users with different language proficiency. Also, a lot of data needs to be conveyed from time to time. Thus UI design needs to be on point.	9	7	63	· Test designs before deployment · Visit the farms to understand the best-practices and use conditions.
System Constraints: The lack of good data service in farms makes it difficult to accurately get the location.	7	5	35	· GPS vs Zip Code accuracy analysis. · Analyze frequency of location change · Implement a fail-safe system
NDI/COTS Conflict: COTS/NDI might have interoperability issues. We also need to consider the cost constraints.	9	5	45	· Review the interoperability of different COTS · Consider COTS cost
Scope Constraint: The multifaceted requirements might evolve and change with the development of the system.	8	4	32	· Prioritized win conditions and expect commitment from the clients. · Expect the client to have negotiable outlook.

6. NDI/NCS Interoperability Analysis

6.1 Introduction

6.1.1 COTS / GOTS / ROTS / Open Source / NCS

Table 10: NDI Products Listing

COTS	Type	Links
Laravel, ASP .NET, AngularJS	Application Framework	
C#, C++, Java	Programming Language	
Twilio, Trumpia, Nexmo, SMSGlobal	Programmatic SMS Messaging	Twilio.com trumpia.com Nexmo.com smsglobal.com
OpenWeatherMap, Weather Underground, Yahoo Weather API, DarkSky	Programmatic fetching of working conditions	openweathermap.org wunderground.com developer.yahoo.com/weather darksky.net

Table 11: Weather API Evaluation Criteria

Evaluation Criteria for Weather API	
Criteria	**Weight**
Update Latency	25
Has Forecast	15

Cost	30
Accepted inputs	5
Output format	10
Comments	15
Total	**100**

Table 12: Comparison of Weather APIs

Programmatic Fetching of Temperature Data						
Service	**Update Latency**	**Has Forecast**	**Cost**	**Accepted inputs**	**Output format**	**Comments**
OpenWeatherMap	Free and $40/mo: < 2 hours (95% LOS) $470/mo: < 10 min (99.5% LOS)	Yes Includes hourly and daily	Free: 60 cpm $40/month: 600 cpm • more	City name, City ID, Lat/Long, Zip Code	JSON, XML, or HTML	Open source. Created to emulate WikiPedia and OpenMaps
Wunderground **"Hyperlocal"**	15 minutes Includes some real time data sources	Yes Including hourly, daily (3-day and 10-day)	Free: 500 cpd, 10 cpm $20/month: 5,000 cpd, 100 cpm • more	City name, Zip Code, Lati/Long, Airport Code, PWS (personal weather station) id, AutoIP address location, IP address	JSON or XML and GIF, PNG or SWF	**Supports multiple languages** (80+)
Yahoo Weather	Unknown	Yes	2,000 cpd	City Name, Zip Code	JSON or XML	Minimal documentation and complicated query structure

DarkSky **"Hyperlocal"**	< 1 hour Includes some real time data sources	Yes Including min-by-min, hourly, daily, weekly, and monthly	1,000 cpd + 0.0001 per forecast after that = 40 cents for 5,000 cpd	Lat/Long for hyperlocal weather	JSON Has wrappers for C#, C++, Java, Javascript, and more	Recently integrated with forecast.io **Supports multiple languages** Tool of choice for many developers

Table 13: Scoring of Weather APIs

Service	Update Latency	Has Forecast	Cost	Accepted inputs	Output format	Comments	Weighted Score
OpenWeatherMap	80	80	92	95	95	85	**86.6**
Wunderground "Hyperlocal"	95	95	80	100	100	90	**90.5**
Yahoo Weather	20	80	90	80	95	20	**60.5**
DarkSky "Hyperlocal"	95	100	100	60	97	100	**96.45**

Table 14: Comparison of SMS APIs

Programmatic SMS Messaging				
Service	**Send + Receive**	**Cost**	**24/7 Support**	**Comments**

Twilio	Yes	$1/mo for #, $0.0075 per SMS (charges for inbound & outbound) $1 = 133 texts	Yes with purchasable plans No guaranteed response time for free Plans from $500-$5000	Used by companies like Uber, Cocacola, and Nordstrom
Trumpia	Yes	Setup costs / monthly costs $0.008 per **outgoing** SMS if pre-pay for 100k texts $0.007 per **outgoing** text if pre-pay for 1m texts $1 = 125-143 texts	Yes Free plan response within 4 *business* hours Premium plan response within 2 *business* hours	Used by various large companies, including Amazon, LinkedIn, Microsoft, eBay, Cocacola, Google, and more
Nexmo	Yes	$0.75/mo for #, $0.0063 per **outgoing** SMS $1 = 159 texts	Yes 2h - 6h guaranteed response time for free 1 support plan for $5000	Owned by Vonage Used by Alibaba and predominantly other European and Asian companies
SMSGlobal	No details	$0.03 - $0.045 per SMS $1 = 22-33 texts	No mention	Used by Microsoft, Budget, Samsung, and IBM Website has significant issues

6.1.2 Connectors

<< Identify the connector, for example
- "In this project, we use PHP/MySQL Connector to enable the PHP web application to retrieve and query data from the database". >>

6.1.3 Legacy System

<< Identify the connector, for example

- "In this project, the development system has to be able to interoperate and works well with "BusinessWorks" version 5.2, which is a software system that the client is currently using." >>

6.2 Evaluation Summary

<< Summarize the final selection of your interoperable NDI/NCS, its usage and its comment. Example can be found in ICSM EPG> Task: Analyze NDI Interoperability for NDI / NCS project. >>

Table 15: NDI Evaluation

NDI	Usages	Comments