Oregon State University
College of Engineering

CS Capstone Design Document

February 11, 2020

BLAMO Design Document

Prepared for

Oregon State University School of Civil and Construction Engineering

Matt Evans _____ Signature _____ Date

Prepared by

Group 36
BLAMO

James Trotter _____ Signature _____ Date

Evan Amaya _____ Signature _____ Date

Sean Spink _____ Signature _____ Date

Alex Smith _____ Signature _____ Date

Abstract

This document outlines the technologies chosen to be used in the development of the BLAMO project for use by the school of civil and construction engineering at OSU. The technologies are structured alongside the relevant design concerns, client concerns, design rationale, structure, intended audience and context in relation to the project as a whole. Additionally, it composes a road map that includes the projected timeline, and phases of development.

CONTENTS

1 INTRODUCTION

1.1 Purpose

This document is written with the intent of defining the technologies used in the development of the BLAMO project. The technologies are outlined alongside the relevant design concern, client concerns, justification for selection, design rationale, and context in relation to the project as a whole to serve as a point of reference.

1.2 Scope

This document is written for the specific application of filling the requirements and needs as outlined by the Oregon State University School of Civil and Construction Engineering, and establishes the technologies that will be used with the intent of filling those needs. The application serves to expedite the current borehole logging process through a mobile application that facilitates remote data recording and generates a visual summary output that is functionally identical to that of gINT - the current industry standard for borehole logging.

1.3 Context

The BLAMO project is being developed for the Oregon State University School of Civil and Construction Engineering through Matt Evans by undergraduate students James Trotter, Evan Amaya, Sean Spink and Alexander Smith as part of the Capstone course. The project will be developed for use by Matt Evans and associated parties in the field, and the developers will be assessed by the Capstone professors based on their performance.

1.4 Influences for the Creation of this Documentation

The preparation of this document is directed and facilitated by collaboration with our client and requirements set forth for the project's completion. This provides an initial point of reference and foundation for the future of the project. The other influences that impact the design and implementation of this project are time constraints of the capstone timeline, client asks and work flow choices, as well as the ability to maintain the project beyond the Capstone course. All of these factors require us to be well-documented and efficient with our implementation for the sake of our client.

1.5 Intended Audience

This document is intended for an audience compiled of members of the field of Geology, Oregon State Geology professionals, and Capstone instructors who will be evaluating the design process for a grade.

2 GLOSSARY

1) BLAMO - Borehole Logging Application Made for Oregon
2) gINT - Geotechnical and Geoenvironmental Software for processing borehole data.
3) LaTeX - Document preparation system centered around plaintext.
4) MikTeX - Windows LaTeX compiler.
5) TeXLive - UNIX LaTeX compiler.
6) MacTeX - MacOS LaTeX compiler.
7) ESRI - Environemntal Systems Research Institute.
8) ArcGIS - Software for processing geospatial information.
9) UI-Framework - Set of software tools for developing graphical user interfaces (GUI).
10) GUI - Everything the user sees and interacts within a piece of software.

3 Programming Language

3.1 Overview

A large design decision that any programming project faces is what programming language to use. There are a lot of native Android programming languages to choose from, but the state of current technology has a divide between multi-platform support. Some suggest it may be easier to build for one platform and then translate code to another (iOS in this case), however we would like to leave this project in an easily adjustable state for the future. A solution to the multi-platform conundrum is Dart, a language that works with the Flutter framework to provide a single code-base for cross platform implementation. Having a single code-base makes it easier for people who are not programming-savvy to make adjustments if they need to. Using Dart was compared to the choice of Java, Kotlin, or C-Sharp, and was chosen due to its near native performance, and cross-platform support.

3.2 Client Concerns

The client has a concern about the future of the project being left in the hands of non-CS related students and professionals. Having a single code-base to work from would reduce both learning curves and complexity. The proprietor of the project would only have to understand a single language, as opposed to the intricacies of two separate languages.

3.3 Rationale

Choosing the right programming language will expedite production of the BLAMO App and reduce the complexity of the code base. This is important to the future of the project, as our client wishes to take this application to an audience beyond the department of geology at OSU. Since the goal is to get the application outside of internal use, having a single code-base will allow for quicker patches, easier understanding, and less micro-management.

3.4 Project Relation

Due to this project being new, having a simple, adaptable, and powerful language will keep possibilities open for the future of development.

4 UI Framework

4.1 Overview

The UI framework will be at the forefront of every user's experience. Since the target market for the BLAMO application is a non-technology oriented field it is crucial to have a UI that is concise, clear, and coherent. As such, Flutter will be used to provide an understandable user interface, as it is important that we have a framework that has cross-platform support, consistency across platforms, and quick performance. Lastly, the UI Framework is going to be doing a lot of the heavy graphical lifting, so having modules that are relevant to displaying geological data is important.

4.2 Client Concerns

To bring the BLAMO project to an audience outside OSU, the client would like the application to look both impressive and comprehensive. Flutter will allow us to provide a clear and coherent UI to the users.

4.3 Rationale

When targeting an application to an audience outside of the institution it is designed in, there are things that become a little more important than when it is exclusive to that institution. When deciding on a framework it came mostly down to language compatibility, features, and native performance. The compared UI frameworks were Xamarin, Flutter, and TornadoFX. Flutter was the best fit for the BLAMO project because it supported easy and consistent cross-platform support with plenty of documentation.

4.4 Project Relation

There are other applications on the market similar to the BLAMO app, so competing with those applications is something that implies not only that the application needs to be functional, but that it also needs to be attractive. Having a strong UI framework with appealing modules and tools will allow for us to create an environment that is enjoyable to work in.

5 Android File Systems

5.1 Overview

The BLAMO app requires a more reliable and permanent data storage solution. There are two file options within the Android operating system, external and internal file systems. Internal file systems are a more secure memory space, and would be exclusive to the application. External file systems are less secure, but provide a more accessible data space. It is the opinion of the developers that the external file space would be the best fit for the BLAMO app.

5.2 Client Concerns

Security was seemingly not a primary concern of the client. However, data integrity seems to be a more crucial concern.

5.3 Rationale

Using external file space ensures two things for the BLAMO app. The external file space will allow for data access regardless of if the devices functionality has been compromised which allows ease of mind for the users in the field. Additionally, using the external file system also allows data to be shared across other applications if other programs wish to use the data produced by the BLAMO app.

5.4 Project Relation

Choosing an appropriate file system will allow for data integrity and ease of mind for every user. The steps for data recovery in case of a device breaking can either be posted online or listed in the documentation which is something a lot of more established programs don't like to discuss. The file system is where the data logs from the user will be stored, in addition to any other meta-data for conversion into PDF or ArcGIS-readable file formats.

6 iOS Port

6.1 Overview

When developing a mobile application it is imperative to make a decision on what platform you will develop on. This is a required point of design because if you choose to develop using a specific language or platform it may not be easy to port your code and designs over and in some cases you won't be able to at all. So instead of pushing the can down the road it is imperative to make the decision in the design stages.

6.2 Client Concerns

The client has not made this design decision a top priority for the project. However, having our project be as accessible as possible to any engineer or scientist looking to utilize the tool is important. Eventually, this project will be usable outside of the school of Civil and Construction Engineering at OSU. Therefore, optimal accessibility and usability is a must.

6.3 Rationale

Porting can prove to be difficult for development on a product. In many cases a company will outsource the porting of their application. This results in the secondary company having to work with what resources they have on the ported platform and can result in the secondary company completely building the application from scratch. This includes redesigning all UI and UX aspects and can cause the applications to be acutely different in severe ways. Flutter will allow for a focus on the requirements and features of the application while periodically testing in both platforms for a well-rounded development strategy.

6.4 Project Relation

This aspect relates to the rest of the project because it is intertwined with how we develop the project. The project language, tools, and available resources that we will be able to utilize is contingent on our decision about what platform to use and what porting we are going to execute. Flutter enables us to focus our time and effort into utilizing one set of tools and features that will provide the best outcome and successfully build the product that the client has asked. On top of this we will save time and effort being able to concurrently test our product on both platforms and allow for a more robust system.

7 Cloud Storage Platform

7.1 Overview

Data storage and data safety are a primary concern for our product. The products features are to improve and ease the burden off of engineers handling the data that they recorded out in the field. With a cloud based storage platform, we can guarantee the user's data will be uploaded, never lost, and accessible anywhere with an internet connection.

7.2 Client Concerns

The client requested that any data that is created on the mobile application should be backed up to the cloud. If there was no direct upload to a cloud then the user would have to go through an extra step of uploading their own data log when they got back to the office. Considering that this project is to expedite the process of engineer data logging and to reduce errors, it would make sense to have the application automatically upload to a cloud storage when possible.

7.3 Rationale

The chosen cloud platform for this project is Box. The reason Box was chosen was mainly due to resource requirements for this project. This is an educationally driven project and the available storage for Box is bigger than that of any other cloud service storage offers for free. On top of that the Box account that we are going to use is contained within the OSU domain. This allows us to not worry about the security of the data because OSU handles all access restrictions and login requirements. Lastly, our client is managing this project after we hand it off and we take into account making the product easier to manage and improve upon with tools that are already familiar to the client.

7.4 Project Relation

This platform will be the second step in the data pipe-lining process for our project. The main purpose will be to receive PDF's that are generated by a user on the mobile application. This cloud location will serve as the main data storage for the application. In result, it will be the main referencing point for any engineer or scientist that wants to look up completed forums from a borehole. It will also allow for easy service to service connection to extract the PDF into the next step of processing the data while never worrying about the data being lost.

8 UX/UI Design

8.1 Overview

With any application it is important to create a design that is easy to use and efficient to use, while still giving the user enough freedom to accomplish their desired task. The tools we use to design and develop the user experience and user interface are therefore very important to the success of the application. These tools should be focused towards developing UX/UI that is suitable for phones and tablets that use Android. We have not yet determined a version of Android to target, that decision will have to balance the improvements of newer versions while still having the application be available to a large enough share of devices running Android. This project will eventually be ported to iOS after Android focused development has concluded, so any UX/UI design choices should take iOS users into account.

8.2 Client Concerns

The client needs a replacement for handwritten forms, any UI/UX we produce should thus be an improvement over that format.

8.3 Rationale

The users of this application will have experience in borehole logging and should have experience with mobile device interfaces, so our design should give them plenty of freedom to accomplish their tasks.

8.4 Project Relation

The user interface is the way that users will be able to access the rest of the features outlined in this document.

9 Testing Frameworks

9.1 Overview

Our testing of this project will focus on the background processes of this application and ensuring that proper outputs are reached for given inputs. The main purpose of this program is to produce drill log documents, as such the testing will focus on the flow of information from the user to the final document output. The majority of this data manipulation will take place locally on the mobile device. Since our initial development will be focused on Android, we will rely on available libraries to create Unit, Component, and Integration automated tests. Any testing performed on elements outside of this flow will likely be performed manually.

9.2 Client Concerns

The client wants an application that is free from bugs.

9.3 Rationale

The majority of our tests will be automated to improve the speed at which we can identify bugs and resolve them. Additionally, these tests will focus on the heart of program, where most of the input and output operations are being performed.

9.4 Project Relation

Testing will be performed on all aspects of this application where it is deemed necessary, this primarily includes any data manipulation performed on the mobile device.

10 gINT Integration

10.1 Overview

One of the major pieces of functionality outlined for our project is to create a visual output that functionally matches that of gINT, the current industry standard for processing borehole logs. Our solution to this is to compile an output file using LaTeX and borehole data that is functionally similar to that of gINT. This solution was compared to the alternative option of automating gINT input which, if possible, would require a large amount of time and would produce compatibility issues in the long run due to the proprietary nature of gINT

10.2 Client Concerns

The client has stated that if gINT compatibility is too much of an issue, alternative avenues of output generation are acceptable.

10.3 Rationale

gINT's primary method of usage as a closed-source professional desktop application is to process borehole input into a visual output. It's primary method of input is through mouse and keyboard. Automating this input would involve creating a method of programmatically simulating mouse and keyboard input within gINT, which would take a large amount of time that could be better spent circumventing gINT. Similarly, this process would also include connecting the mobile application to a desktop machine. Part of our goal is to expedite the process of creating a summary page from raw data while hiding as much of the process from the user as possible. Requiring connection to a desktop machine violates this rationale, whereas generating output within the app is in line with it.

10.4 Project Relation

Creating this output is the cornerstone of our project, and everything is designed with the end goal of creating this output as easily as possible. Working directly in gINT would contradict our rationale and thus other aspects of the project must accommodate an independent way of output generation, such as LaTeX.

11 Visual Output

11.1 Overview

Since it may be prudent to circumvent gINT, we will need a method of programmatically compiling borehole information stored in the database into a one-page summary, with information grouped together into columns under headers and formatted accordingly. The best option for this would be to curate a .tex file with the borehole's information in a custom LaTeX template.

11.2 Client Concerns

The client has stated that if gINT compatibility is too much of an issue, alternative avenues of output generation are acceptable.

11.3 Rationale

A LaTeX document is compiled from a base .tex file. Additionally, LaTeX has a number of tags for creating, combining and editing tables, rows and columns, in addition to its extensive functionality for controlling text attributes such as font, size, margins, etc. Creating the .tex file would be simple to implement using basic file i/o and inserting formatted strings of data into a preexisting .tex template. In theory, all of this should be able to be executed in-app in the background, adhering to our goal of expediting the process while hiding it from the user.

11.4 Project Relation

The back-end of the application will need to be compatible with one of many open-source LaTeX compilers available, such as MikTeX, TeXLive or MacTeX.

12 ArcGIS Integration

12.1 Overview

If the project progresses accordingly, we may be looking to implement automatic integration into an ArcGIS geospatial database. ArcGIS is a program developed by ESRI "for organizations to create, manage, share and analyze spatial data. It consists of server components, mobile and desktop applications, and developer tools." Of these tools, ArcGIS allows for data retrieval from a database for geospatial display through SQL or through their Python API. The graduate student project, currently in development, reads .csv output files created from gINT. The Capstone team and the graduate student project have agreed to create a common .csv format with a set of expected parameters that matches that of gINT with some potential project-specific additions.

12.2 Client Concerns

Our client works with graduate students in geospatial studies who currently work on developing and maintaining a current geospatial database of field data collected in the Pacific Northwest. Therefore any methods for GIS data conversion outlined in this document are subject to change based on feedback from the graduate student project.

12.3 Rationale

Direct compatibility with ArcGIS databases will not be necessary for the scope of this project - only compatibility with the graduate student project is required. As such, creating an additional piece of output that is compatible with their project will fulfill the project requirements and expand the BLAMO application's utility.

12.4 Project Relation

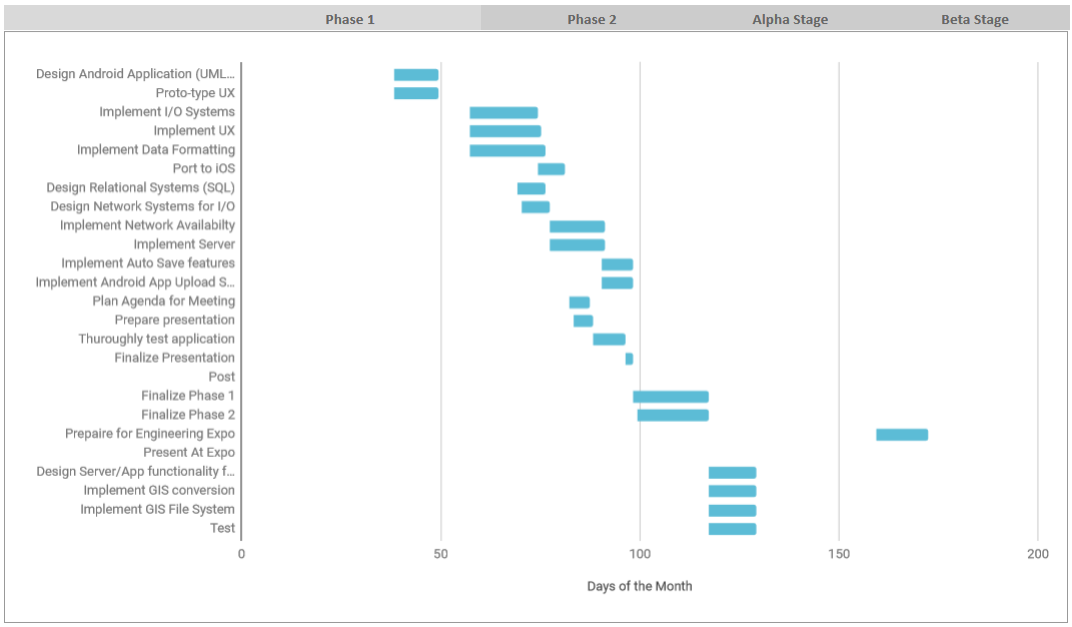
Our project's stretch goal is to work with the graduate students to develop a method of programmatically uploading the collected borehole data from our database to theirs for geospatial display and analysis. Therefore, its completion is dependent on the rest of the program's, and should be considered a modular addition.

13 Work-flow Overview

Our current development cycle will work in stages of design, implementation, testing, and roll out:

- Design will include methods such as prototyping, UML design, and writing pseudo code.
- Implementation will be executed by assigning segments to the team members to work on. The team members will communicate with other members to make sure each module flows with the other modules.
- Testing will include some automated testing that will be written in conjunction with the Application code base in an effort to follow test driven development.
- Rollout will have team members looking over each others code to make sure it is coherent and bug free (to the best of our abilities)

TASK NAME	START DATE	DAY OF MONTH*	END DATE	DURATION* (WORK DAYS)	DAYS COMPLETE*	DAYS REMAINING*	TEAM MEMBER	PERCENT COMPLETE
Phase 1								
Design Android Application (UML, Tasks)	1/1	1	1/12	11	11	38.30258171	Group	100%
Proto-type UX	1/1	1	1/12	11	8.8	38.30258171	Group	80%
Implement I/O Systems	1/20	20	2/6	17	10.2	57.30258171	TBA	60%
Implement UX	1/20	20	2/7	18	7.2	57.30258171	TBA	40%
Implement Data Formatting	1/20	20	2/8	19	3.8	57.30258171	TBA	20%
Port to iOS	2/6	6	2/13	7	0	74.30258171	TBA	0%
Phase 2								
Design Relational Systems (SQL)	2/1	1	2/8	7	7	69.30258171	Group	100%
Design Network Systems for I/O	2/2	2	2/9	7	5.6	70.30258171	Group	80%
Implement Network Availabilty	2/9	9	2/23	14	8.4	77.30258171	TBA	60%
Implement Server	2/9	9	2/23	14	5.6	77.30258171	TBA	40%
Implement Auto Save features	2/22	22	3/1	8	1.6	90.30258171	TBA	20%
Implement Android App Upload Settings	2/22	22	3/1	8	0	90.30258171	TBA	0%
Alpha Stage								
Plan Agenda for Meeting	2/14	14	2/19	5	5	82.30258171	Group	100%
Prepare presentation	2/15	15	2/20	5	4	83.30258171	Group	80%
Thuroughly test application	2/20	20	2/28	8	4.8	88.30258171	Group	60%
Finalize Presentation	2/28	28	3/1	2	0.8	96.30258171	Group	40%
Post	03/2020	1	3/1	0	0	98.30258171	Group	20%
Beta Stage								
Finalize Phase 1	3/1	1	3/20	19	19	98.30258171	Group	100%
Finalize Phase 2	3/2	2	3/20	18	14.4	99.30258171	TBA	80%
Prepaire for Engineering Expo	5/1	1	5/14	13	7.8	159.3025817	TBA	60%
Present At Expo	5/15	15	5/15	0	0	173.3025817	TBA	0%
Phase 3 (Stretch)								
Design Server/App functionality for GIS	3/20	20	4/1	12	12	117.3025817	Group	100%
Implement GIS conversion	3/20	20	4/1	12	9.6	117.3025817	TBA	80%
Implement GIS File System	3/20	20	4/1	12	7.2	117.3025817	TBA	60%
Test	3/20	20	4/1	12	0	117.3025817	TBA	0%


Phase 1
Phase 2
Alpha Stage
Beta Stage
Design Android Application (UML...
Proto-type UX
Implement I/O Systems
Implement UX
Implement Data Formatting
Port to iOS
Design Relational Systems (SQL)
Design Network Systems for I/O
Implement Network Availabilty
Implement Server
Implement Auto Save features
Implement Android App Upload S...
Plan Agenda for Meeting
Prepare presentation
Thuroughly test application
Finalize Presentation
Post
Finalize Phase 1
Finalize Phase 2
Prepaire for Engineering Expo
Present At Expo
Design Server/App functionality f...
Implement GIS conversion
Implement GIS File System
Test
0
50
100
150
200
Days of the Month