
CrowdControl

Senior Design Final Documentation

BowTaps

Johnnathon Ackerman

Daniel Andrus

Charles Bonn

Evan Hammer

Joseph Mowry

January 18, 2016

Contents

Title	i
Contents	vi
List of Figures	vii
List of Tables	ix
List of Algorithms	xi
Overview Statements	xiii
0.1 Mission Statement	xiii
0.2 Elevator Pitch	xiii
Document Preparation and Updates	xv
1 Overview and concept of operations	1
1.1 Team Members and Team Name	1
1.2 Client	1
1.3 Project	1
1.3.1 Purpose of the System	1
1.4 Business Need	1
1.5 Deliverables	1
1.6 System Description	1
1.6.1 Integrated Group Messaging	2
1.6.2 GPS Location services	2
1.6.3 Group Management Features	2
1.6.4 Suggestions	2
1.7 System Overview and Diagram	2
1.8 Technologies Overview	2
1.8.1 Google Play Services	2
1.8.2 Apple Map Features	3
1.8.3 Parse	3
2 User Stories, Requirements, and Product Backlog	5
2.1 Overview	5
2.2 User Stories	5
2.2.1 User Story #1	5
2.2.2 User Story #2	5
2.2.3 User Story #3	5
2.2.4 User Story #4	5
2.2.5 User Story #5	5
2.2.6 User Story #6	6
2.2.7 User Story #7	6

2.2.8	User Story #8	6
2.2.9	User Story #9	6
2.2.10	User Story #10	6
2.3	Requirements and Design Constraints	6
2.3.1	System Requirements	6
2.3.2	Network Requirements	6
2.3.3	Development Environment Requirements	6
2.3.4	Project Management Methodology	6
2.4	Specifications	6
2.5	Product Backlog	6
2.6	Research or Proof of Concept Results	7
2.7	Supporting Material	7
3	Project Overview	9
3.1	Team Member's Roles	9
3.2	Project Management Approach	10
3.3	Stakeholder Information	10
3.3.1	Customer or End User (Product Owner)	10
3.3.2	Management or Instructor (Scrum Master)	10
3.3.3	Investors	10
3.3.4	Developers –Testers	10
3.4	Budget	10
3.5	Intellectual Property and Licensing	10
3.6	Sprint Overview	10
3.7	Terminology and Acronyms	11
3.8	Sprint Schedule	11
3.9	Timeline	11
3.10	Backlogs	11
3.11	Burndown Charts	11
3.12	Development Environment	11
3.13	Development IDE and Tools	11
3.14	Source Control	11
3.15	Dependencies	11
3.16	Build Environment	11
3.17	Development Machine Setup	11
4	Design and Implementation	13
4.1	Architecture and System Design	13
4.1.1	Design Selection	14
4.1.2	Data Structures and Algorithms	14
4.1.3	Data Flow	14
4.1.4	Communications	14
4.1.5	Classes	14
4.1.6	UML	14
4.1.7	GUI	14
4.1.8	MVVM, etc	14
4.2	Major Component #1	14
4.2.1	Technologies Used	14
4.2.2	Component Overview	14
4.2.3	Phase Overview	14
4.2.4	Architecture Diagram	14
4.2.5	Data Flow Diagram	14
4.2.6	Design Details	14
4.3	Major Component #2	15
4.3.1	Technologies Used	15

4.3.2	Component Overview	15
4.3.3	Phase Overview	15
4.3.4	Architecture Diagram	15
4.3.5	Data Flow Diagram	15
4.3.6	Design Details	15
4.4	Major Component #3	15
4.4.1	Technologies Used	15
4.4.2	Component Overview	15
4.4.3	Phase Overview	16
4.4.4	Architecture Diagram	16
4.4.5	Data Flow Diagram	16
4.4.6	Design Details	16
5	System and Unit Testing	17
5.1	Overview	17
5.2	Dependencies	17
5.3	Test Setup and Execution	17
5.4	System Testing	17
5.5	System Integration Analysis	17
5.6	Risk Analysis	17
5.6.1	Risk Mitigation	17
5.7	Successes, Issues and Problems	17
5.7.1	Changes to the Backlog	17
6	Prototypes	19
6.1	Sprint 1 Prototype	19
6.1.1	Deliverable	19
6.1.2	Backlog	19
6.1.3	Success/Fail	19
6.2	Sprint 2 Prototype	19
6.2.1	Deliverable	19
6.2.2	Backlog	19
6.2.3	Success/Fail	19
6.3	Sprint 3 Prototype	19
6.3.1	Deliverable	19
6.3.2	Backlog	19
6.3.3	Success/Fail	19
6.4	Sprint 4 Prototype	19
6.4.1	Deliverable	19
6.4.2	Backlog	19
6.4.3	Success/Fail	19
6.5	Sprint 5 Prototype	19
6.5.1	Deliverable	19
6.5.2	Backlog	19
6.5.3	Success/Fail	20
7	Release – Setup – Deployment	21
7.1	Deployment Information and Dependencies	21
7.2	Setup Information	21
7.3	System Versioning Information	21
8	User Documentation	23
8.1	User Guide	23
8.2	Installation Guide	23
8.3	Programmer Manual	23

9	Class Index	25
9.1	Class List	25
10	Class Documentation	27
10.1	Poly Class Reference	27
10.1.1	Constructor & Destructor Documentation	27
10.1.2	Member Function Documentation	27
11	Business Plan	29
11.1	Business Model	29
11.2	Market and Competition	29
11.3	Regulatory environment	29
11.4	Intellectual Property and Freedom to Operate	29
11.5	Management Team and Advisors	29
11.6	Sources and Uses of Capital	29
11.7	Financial Statements	29
11.8	Metrics and Milestones	29
11.9	Exit Plan	29
12	Experimental Log	31
13	Research Results	33
13.1	Result 1	33
13.2	Result 2	33
13.3	Conclusions	33
13.4	Further work	33
	Bibliography	35
	Software Agreement	SA-1
A	Product Description	A-1
B	Publications	B-1
C	Sprint Reports	C-1
1	Sprint Report #1	C-1
2	Sprint Report #2	C-1
3	Sprint Report #3	C-1
4	Sprint Report	C-1
D	Industrial Experience and Resumes	D-1
1	Resumes	D-1
2	ABET: Industrial Experience Reports	D-1
2.1	Name1	D-1
2.2	Name2	D-1
2.3	Name3	D-1
E	Acknowledgment	E-1
F	Supporting Materials	F-1
L^AT_EX	Example	BM-1
1	Introduction	BM-1
2	Ordinary Text	BM-1
3	Displayed Text	BM-2
4	Build process	BM-2

List of Figures

1.1 Basic System Flow Diagram	2
---	---

List of Tables

List of Algorithms

1	Calculate $y = x^n$	13
---	-------------------------------	----

Overview Statements

0.1 Mission Statement

Our mission at BowTaps is to develop innovative mobile software applications to provide solutions to inconveniences that trouble the everyday user. With our software we plan on changing the mobile environment by creating applications with easy to use and intuitive interfaces with reliable services for everyday use.

0.2 Elevator Pitch

Our company, Bowtaps, is developing an iPhone/Android app to help young adults and event-goers stay in contact with friends while in loud and crowded places using group messaging and GPS features.

Our product, Crowd Control, is designed to become an essential element for groups looking to go out together by providing both powerful group-management tools and interesting nearby outing suggestions, such as local events, concerts, and pub crawls.

We will work with local businesses and event planners to sponsor these suggestions in order generate content for our users, visibility for our sponsors, and revenue for ourselves.

We plan to release the app for free in early-to-mid summer of 2016.

Document Preparation and Updates

Current Version [X.X.X]

Prepared By:
Charles Bonn
Johnathon Ackerman
Daniel Andrus
Evan Hammer
Joseph Mowry

Revision History

<i>Date</i>	<i>Author</i>	<i>Version</i>	<i>Comments</i>
<i>1/8/14</i>	<i>Charles Bonn</i>	<i>1.0.0</i>	<i>Refactor to new design document</i>

1

Overview and concept of operations

1.1 Team Members and Team Name

The team name is BowTaps. Bow taps currently consists of the members Charles Bonn, Johnathon Ackerman, Daniel Andrus, Evan Hammer, and Joesph Mowry.

1.2 Client

A description of the client or customer.

1.3 Project

A high level description of the project.

1.3.1 Purpose of the System

What is the purpose of the system or product?

1.4 Business Need

Use this section to define what business need exist and how this software will meet and/or exceed that business need.

1.5 Deliverables

Provide a complete description of the client requested deliverables. This section should be the section your software contract references.

1.6 System Description

Crowd Control is a mobile application designed to ease the experience of going out though the implimentation of integrated group messaging, GPS tracking and group management features. Along with the features to manage your group at the event Crowd Control also gives suggestions of local events, restraunts and attraction. This allows the group to continue even when the next item on the agenda is a mystry.

Even though Crowd Control is designed for the party sceen and people going out to events, it uses can be expanded to fit more purposes. Crowd Control can be used to help manage any kind of group at an event such as church groups or school field trips.

1.6.1 Integrated Group Messaging

Integrated group messaging is an important feature of Crowd Control. Integrated group messaging allows for communication between cross platform, different phone brands, and different carriers. This allows for seamless communication between users without the issues associated with messaging such as messages not using the same format, messages not going to all recipients, and messages with users in the group that you do not want to have your personal information.

1.6.2 GPS Location services

GPS allows for tracking of members in the group on a local map of the area. With this feature you will be able to keep track of anyone in the group off of their last GPS check in. This is useful to help locate members of the group that maybe lost or unable to be located. This feature will have the option of being able to opt out when the user does not want to have their location known to the group. When the user's battery is low it will allow for the check in period to be extended or turned off to save battery life.

1.6.3 Group Management Features

The group management features allow for information to be shared with the group. A group management menu will allow for a group agenda to be posted as well as updates when the agenda changes. With the GPS features it will allow for the group leader to set way points for the group.

1.6.4 Suggestions

Suggestions are both a plus for the user and our way of making a monetary development. Suggestions are sponsored by local businesses in the form of an ad. Although these are not traditional ads, they are in the form of local points of interest such as restaurants, bars, amusement parks, or bowling alleys. The possibilities are endless. With the suggestion method it will allow for our users to have helpful suggestions of places for their group to attend as well as exposure for the local businesses that are sponsoring Crowd Control.

1.7 System Overview and Diagram

The basic overview of Crowd Control can be seen in the diagram below. See Figure 1.1. Crowd Control will be using a model view controller design structure. With the model view controller design method we are able to abstract the user interface from the control structures that will communicate with the third party services such as Parse, Google Play services, or Apple Map Features. The model of each respective operating system (Android or iOS) will be able to communicate with the respective mapping feature (Google Play Services or Apple Map Features). While both models will be able to communicate with Parse, our backend server. Though Parse, using their features, will be able to connect user profiles to their facebook and twitter accounts for faster login.

Figure 1.1: Basic System Flow Diagram

1.8 Technologies Overview

Some technologies used in the creation of Crowd Control are Google Play Services, Apple Map Features, and Parse.

1.8.1 Google Play Services

1.8.1.a Description

Google Play Services contains the native android API for mapping features. With this it allows for communication between a map and your gps location along with other mapping features.

REFERENCE LINK: <https://developers.google.com/android/guides/setup>

1.8.1.b Usage

Google Play Services will be used on the Android device as the default map. We chose to go with Google Play services to give android users a more native feel when it comes to using the mapping features. This allows for a less intrusive feel when it comes to using Crowd Control. This will be used for displaying your location on a map, displaying other users in your group on a map, and displaying event suggestions on the map.

1.8.2 Apple Map Features

1.8.2.a Description

Apple Map Features is the native iOS API for mapping features. With this it allows for communication between a map and your gps location along with other mapping features.

REFERENCE LINK: <https://developer.apple.com/maps/>

1.8.2.b Usage

Apple Map Features will be used on the iOS device as the default map. We chose to go with Apple Map Features to give iOS users a more native feel when it comes to using the mapping features. This allows for a less intrusive feel when it comes to using Crowd Control. This will be used for displaying your location on a map, displaying other users in your group on a map, and displaying event suggestions on the map.

1.8.3 Parse

1.8.3.a Description

Parse is our backend database. It allows us to save information that is needed along with giving us a way to connect to both facebook and twitter.

REFERENCE LINK: <http://parse.com/>

1.8.3.b Usage

Parse will be used to save information, group information, and advertisement information. It will be the main communication between devices and past user information.

2

User Stories, Requirements, and Product Backlog

2.1 Overview

The overview should take the form of an executive summary. Give the reader a feel for the purpose of the document, what is contained in the document, and an idea of the purpose for the system or product.

The user stories are provided by the stakeholders. You will create the backlogs and the requirements, and document here. This chapter should contain details about each of the requirements and how the requirements are or will be satisfied in the design and implementation of the system.

Below: list, describe, and define the requirements in this chapter. There could be any number of sub-sections to help provide the necessary level of detail.

2.2 User Stories

2.2.1 User Story #1

As a user i want to be able to join a group.

2.2.1.a User Story #1 Breakdown

As a user i want the ability to join a group. Group joining options would be from a list or from an invite from a user.

2.2.2 User Story #2

As a user i want the ability to track locations of other members in the group.

2.2.2.a User Story #2 Breakdown

2.2.3 User Story #3

As a user i want post agenda for the group.

2.2.4 User Story #4

As a user i want to i want the ability to look for local groups

2.2.5 User Story #5

As a user i want the ability to have suggestions of local activities.

2.2.6 User Story #6

As a user i want the ability to leave a group.

2.2.7 User Story #7

As a user i want the ability to have a list of local groups.

2.2.8 User Story #8

As a user i want the ability to login.

2.2.9 User Story #9

As a user i would like to message other members of the group.

2.2.10 User Story #10

As a user i would like my information protected.

2.3 Requirements and Design Constraints

Use this section to discuss what requirements exist that deal with meeting the business need. These requirements might equate to design constraints which can take the form of system, network, and/or user constraints. Examples: Windows Server only, iOS only, slow network constraints, or no offline, local storage capabilities.

2.3.1 System Requirements

What are they? How will they impact the potential design? Are there alternatives?

2.3.2 Network Requirements

What are they?

2.3.3 Development Environment Requirements

What are they? Is the system supposed to be cross-platform?

2.3.4 Project Management Methodology

The stakeholders might restrict how the project implementation will be managed. There may be constraints on when design meetings will take place. There might be restrictions on how often progress reports need to be provided and to whom.

2.4 Specifications

Any specifications that need to be understood? Put it here.

2.5 Product Backlog

The full product backlog should go here. The sprint backlogs are located in the project chapter.

- What system will be used to keep track of the backlogs and sprint status?
- Will all parties have access to the Sprint and Product Backlogs?

- How many Sprints will encompass this particular project?
- How long are the Sprint Cycles?
- Are there restrictions on source control?

2.6 Research or Proof of Concept Results

This section is reserved for the discussion centered on any research that needed to take place before full system design. The research efforts may have led to the need to actually provide a proof of concept for approval by the stakeholders. The proof of concept might even go to the extent of a user interface design or mockups.

2.7 Supporting Material

This document might contain references or supporting material which should be documented and discussed either here if appropriate or more often in the appendices at the end. This material may have been provided by the stakeholders or it may be material garnered from research tasks.

3

Project Overview

This section provides some housekeeping type of information with regard to the team, project, environment, etc.

3.1 Team Member's Roles

Johnnathon Ackerman - Johnnathon is leading the GUI design and implimentation side for the android version of Crowd Control. This entails:

1. Graphical Design
2. Smooth Moving Interfaces
3. Easy to Use and learn layout

Daniel Andrus - Daniel is leading the Gui design ad implimentation for the iOS version of Crowd Control. This entails:

1. Graphical Design
2. Smooth Moving Interfaces
3. Easy to Use and learn layout

Charles Bonn - Charles is leading the database side of Crowd Control. This database is for both iOS and andriod versions. This entails:

1. Creating and managing database qurries
2. Creating Cloud Code to manage database information
3. Database load testing

Charles is also working on future encryption of data going to and from the database.

Evan Hammer - Evan is leading the backend side for the iOS version of Crowd Control. This entails:

1. Creating links from the database to the mobile application
 - (a) Login link
 - (b) Group Join Link
 - (c) Group Member

2. Creating links to Apple maps to the mobile application

Joseph Mowry - Joseph is leading the backend side for the android version of Crowd Control. This entails:

1. Creating links from the database to the mobile application

- (a) Login link
- (b) Group Join Link
- (c) Group Member

2. Creating links to Apple maps to the mobile application

3.2 Project Management Approach

This section will provide an explanation of the basic approach to managing the project. Typically, this would detail how the project will be managed through a given Agile methodology. The sprint length (i.e. 2 weeks) and product backlog ownership and location (ex. Trello) are examples of what will be discussed. An overview of the system used to track sprint tasks, bug or trouble tickets, and user stories would be warranted.

3.3 Stakeholder Information

This section would provide the basic description of all of the stakeholders for the project. Who has an interest in the successful and/or unsuccessful completion of this project?

3.3.1 Customer or End User (Product Owner)

Who? What role will they play in the project? Will this person or group manage and prioritize the product backlog? Who will they interact with on the team to drive product backlog priorities if not done directly?

3.3.2 Management or Instructor (Scrum Master)

Who? What role will they play in the project? Will the Scrum Master drive the Sprint Meetings?

3.3.3 Investors

Are there any? Who? What role will they play?

3.3.4 Developers –Testers

Who? Is there a defined project manager, developer, tester, designer, architect, etc.?

3.4 Budget

Describe the budget for the project including gifted equipment and salaries for people on the project.

3.5 Intellectual Property and Licensing

Describe the IP ownership and issues surrounding IP.

3.6 Sprint Overview

If the system will be implemented in phases, describe those phases/sub-phases (design, implementation, testing, delivery) and the various milestones in this section. This section should also contain a correlation between the phases of development and the associated versioning of the system, i.e. major version, minor version, revision.

All of the Agile decisions are listed here. For example, how do you order your backlog? Did you use planning poker?

3.7 Terminology and Acronyms

Provide a list of terms used in the document that warrant definition. Consider industry or domain specific terms and acronyms as well as system specific.

3.8 Sprint Schedule

The sprint schedule. Can be tables or graphs. This can be a list of dates with the visual representation given below.

3.9 Timeline

Gantt chart or other type of visual representation of the project timeline.

3.10 Backlogs

Place the sprint backlogs here. The product backlog will be in the chapter with the user stories.

3.11 Burndown Charts

Place your burndown charts, team velocity information, etc here.

3.12 Development Environment

The basic purpose for this section is to give a developer all of the necessary information to setup their development environment to run, test, and/or develop.

3.13 Development IDE and Tools

Describe which IDE and provide links to installs and/or reference material.

3.14 Source Control

Which source control system is/was used? How was it setup? How does a developer connect to it?

3.15 Dependencies

Describe all dependencies associated with developing the system.

3.16 Build Environment

How are the packages built? Are there build scripts?

3.17 Development Machine Setup

If warranted, provide a list of steps and details associated with setting up a machine for use by a developer.

4

Design and Implementation

This section is used to describe the design details for each of the major components in the system. Note that this chapter is critical for all tracks. Research tracks would do experimental design here where other tracks would include the engineering design aspects. This section is not brief and requires the necessary detail that can be used by the reader to truly understand the architecture and implementation details without having to dig into the code. Sample algorithm: Algorithm 1. This algorithm environment is automatically placed - meaning it floats. You don't have to worry about placement or numbering.

Algorithm 1 Calculate $y = x^n$

Require: $n \geq 0 \vee x \neq 0$

Ensure: $y = x^n$

```
 $y \leftarrow 1$ 
if  $n < 0$  then
   $X \leftarrow 1/x$ 
   $N \leftarrow -n$ 
else
   $X \leftarrow x$ 
   $N \leftarrow n$ 
end if
while  $N \neq 0$  do
  if  $N$  is even then
     $X \leftarrow X \times X$ 
     $N \leftarrow N/2$ 
  else  $\{N \text{ is odd}\}$ 
     $y \leftarrow y \times X$ 
     $N \leftarrow N - 1$ 
  end if
end while
```

Citations look like [2, 1, 3] and [6, 4, 5]. These are done automatically. Just fill in the database `designrefs.bib` using the same field structure as the other entries. Then `pdflatex` the document, `bibtex` the document and `pdflatex` twice again. The first `pdflatex` creates requests for bibliography entries. The `bibtex` extracts and formats the requested entries. The next `pdflatex` puts them in order and assigns labels. The final `pdflatex` replaces references in the text with the assigned labels. The bibliography is automatically constructed.

4.1 Architecture and System Design

This is where you will place the overall system design or the architecture. This section should be image rich. There is the old phrase *a picture is worth a thousand words*, in this class it could be worth a hundred points (well if you sum up over the entire team). One needs to enter the design and why a particular design has been done.

4.1.1 Design Selection

Failed designs, design ideas, rejected designs here.

4.1.2 Data Structures and Algorithms

Describe the special data structures and any special algorithms.

4.1.3 Data Flow

4.1.4 Communications

4.1.5 Classes

4.1.6 UML

4.1.7 GUI

4.1.8 MVVM, etc

4.2 Major Component #1

4.2.1 Technologies Used

This section provides a list of technologies used for this component. The details for the technologies have already been provided in the Overview section.

4.2.2 Component Overview

This section can take the form of a list of features.

4.2.3 Phase Overview

This is an extension of the Phase Overview above, but specific to this component. It is meant to be basically a brief list with space for marking the phase status.

4.2.4 Architecture Diagram

It is important to build and maintain an architecture diagram. However, it may be that a component is best described visually with a data flow diagram.

4.2.5 Data Flow Diagram

It is important to build and maintain a data flow diagram. However, it may be that a component is best described visually with an architecture diagram.

4.2.6 Design Details

This is where the details are presented and may contain subsections. Here is an example code listing:

```
#include <stdio.h>
#define N 10
/* Block
 * comment */

int main()
{
    int i;
```

```
// Line comment.  
puts("Hello world!");  
  
for (i = 0; i < N; i++)  
{  
    puts("LaTeX is also great for programmers!");  
}  
  
return 0;  
}
```

This code listing is not floating or automatically numbered. If you want auto-numbering, put it in the algorithm environment (not algorithmic however) shown above.

4.3 Major Component #2

4.3.1 Technologies Used

This section provides a list of technologies used for this component. The details for the technologies have already been provided in the Overview section.

4.3.2 Component Overview

This section can take the form of a list of features.

4.3.3 Phase Overview

This is an extension of the Phase Overview above, but specific to this component. It is meant to be basically a brief list with space for marking the phase status.

4.3.4 Architecture Diagram

It is important to build and maintain an architecture diagram. However, it may be that a component is best described visually with a data flow diagram.

4.3.5 Data Flow Diagram

It is important to build and maintain a data flow diagram. However, it may be that a component is best described visually with an architecture diagram.

4.3.6 Design Details

This is where the details are presented and may contain subsections.

4.4 Major Component #3

4.4.1 Technologies Used

This section provides a list of technologies used for this component. The details for the technologies have already been provided in the Overview section.

4.4.2 Component Overview

This section can take the form of a list of features.

4.4.3 Phase Overview

This is an extension of the Phase Overview above, but specific to this component. It is meant to be basically a brief list with space for marking the phase status.

4.4.4 Architecture Diagram

It is important to build and maintain an architecture diagram. However, it may be that a component is best described visually with a data flow diagram.

4.4.5 Data Flow Diagram

It is important to build and maintain a data flow diagram. However, it may be that a component is best described visually with an architecture diagram.

4.4.6 Design Details

This is where the details are presented and may contain subsections.

5

System and Unit Testing

This section describes the approach taken with regard to system and unit testing.

5.1 Overview

Provides a brief overview of the testing approach, testing frameworks, and general how testing is/will be done to provide a measure of success for the system.

Each requirement (user story component) should be tested. A review of objectives and constraints might be needed here.

5.2 Dependencies

Describe the basic dependencies which should include unit testing frameworks and reference material.

5.3 Test Setup and Execution

Describe how test cases were developed, setup, and executed. This section can be extremely involved if a complete list of test cases was warranted for the system. One approach is to list each requirement, module, or component and describe the test.

The unit tests are described here.

5.4 System Testing

5.5 System Integration Analysis

5.6 Risk Analysis

5.6.1 Risk Mitigation

5.7 Successes, Issues and Problems

5.7.1 Changes to the Backlog

6

Prototypes

This chapter is for recording each prototype developed. It is a historical record of what you accomplished in 464/465. This should be organized according to Sprints. It should have the basic description of the sprint deliverable and what was accomplished. Screen shots, photos, captures from video, etc should be used.

6.1 Sprint 1 Prototype

6.1.1 Deliverable

6.1.2 Backlog

6.1.3 Success/Fail

6.2 Sprint 2 Prototype

6.2.1 Deliverable

6.2.2 Backlog

6.2.3 Success/Fail

6.3 Sprint 3 Prototype

6.3.1 Deliverable

6.3.2 Backlog

6.3.3 Success/Fail

6.4 Sprint 4 Prototype

6.4.1 Deliverable

6.4.2 Backlog

6.4.3 Success/Fail

6.5 Sprint 5 Prototype

6.5.1 Deliverable

6.5.2 Backlog

6.5.3 Success/Fail

7

Release – Setup – Deployment

This section should contain any specific subsection regarding specifics in releasing, setup, and/or deployment of the system.

7.1 Deployment Information and Dependencies

Are there dependencies that are not embedded into the system install?

7.2 Setup Information

How is a setup/install built?

7.3 System Versioning Information

How is the system versioned?

8

User Documentation

This section should contain the basis for any end user documentation for the system. End user documentation would cover the basic steps for setup and use of the system. It is likely that the majority of this section would be present in its own document to be delivered to the end user. However, it is recommended the original is contained and maintained in this document.

8.1 User Guide

The source for the user guide can go here. You have some options for how to handle the user docs. If you have some `newpage` commands around the guide then you can just print out those pages. If a different formatting is required, then have the source in a separate file `userguide.tex` and include that file here. That file can also be included into a driver (like the senior design template) which has the client specified formatting. Again, this is a single source approach.

8.2 Installation Guide

8.3 Programmer Manual

9

Class Index

9.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Poly 27

10

Class Documentation

10.1 Poly Class Reference

Public Member Functions

- Poly ()
- ~Poly ()
- int myfunction (int)

10.1.1 Constructor & Destructor Documentation

10.1.1.a Poly::Poly ()

My constructor

10.1.1.b Poly::~~Poly ()

My destructor

10.1.2 Member Function Documentation

10.1.2.a int Poly::myfunction (int *a*)

my own example function fancy new function

new variable

The documentation for this class was generated from the following file:

- hello.cpp

11

Business Plan

- 11.1 Business Model**
- 11.2 Market and Competition**
- 11.3 Regulatory environment**
- 11.4 Intellectual Property and Freedom to Operate**
- 11.5 Management Team and Advisors**
- 11.6 Sources and Uses of Capital**
- 11.7 Financial Statements**
- 11.8 Metrics and Milestones**
- 11.9 Exit Plan**

12

Experimental Log

For research projects one needs to keep a log of all research/lab activities.

10/15/15 Ran modified filter on data sets 1 - 6. Results were ...

10/17/15 Changed tolerance on sensor and collected data. These ...

13

Research Results

This chapter describes the results and conclusions of your research. This would be the final report for a research project.

13.1 Result 1

13.2 Result 2

13.3 Conclusions

13.4 Further work

Bibliography

- [1] R. Arkin. *Governing Lethal Behavior in Autonomous Robots*. Taylor & Francis, 2009.
- [2] Howie Choset, Kevin M. Lynch, Seth Hutchinson, George A Kantor, Wolfram Burgard, Lydia E. Kavraki, and Sebastian Thrun. *Principles of Robot Motion: Theory, Algorithms, and Implementations*. MIT Press, Cambridge, MA, June 2005.
- [3] S. M. LaValle. *Planning Algorithms*. Cambridge University Press, Cambridge, U.K., 2006. Available at <http://planning.cs.uiuc.edu/>.
- [4] V. Lumelsky and A. Stepanov. Path planning strategies for point mobile automation moving amidst unknown obstacles of arbitrary shape. *Algorithmica*, pages 403–430, 1987.
- [5] S.A. NOLFI and D.A. FLOREANO. *Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines*. A Bradford book. A BRADFORD BOOK/THE MIT PRESS, 2000.
- [6] Wikipedia. Asimo — Wikipedia, the free encyclopedia. http://upload.wikimedia.org/wikipedia/commons/thumb/0/05/HONDA_ASIMO.jpg/450px-HONDA_ASIMO.jpg, 2013. [Online; accessed June 23, 2013].

SDSMT SENIOR DESIGN SOFTWARE DEVELOPMENT AGREEMENT

This Software Development Agreement (the “Agreement”) is made between the SDSMT Computer Science Senior Design Team _____,
(“Student Group”)
consisting of team members _____,
(“Student Names”)
and Sponsor _____,
(“Company Name”)
with address: _____.

[Note: Bracketed material is included to suggest content that will vary with each agreement. I STRONGLY SUGGEST THAT THE INSTRUCTOR LOOK AT THE COMPLETED AGREEMENT BEFORE YOU SIGN IT!!]

1 RECITALS

1. Sponsor desires Senior Design Team to develop software [for use in Sponsor’s simulation platform for optical fiber transmissions of digitized video signals] (the ”Field”).
2. Senior Design Teams willing to develop such Software.

NOW, THEREFORE, in consideration of the mutual covenants and promises herein contained, the Team and Sponsor agree as follows:

2 EFFECTIVE DATE

This Agreement shall be effective as of _____ (the “Effective Date”).

3 DEFINITIONS

1. “Software” shall mean [the computer programs in machine readable object code form and any subsequent error corrections or updates supplied to Sponsor by Senior Design Team pursuant to this Agreement.] [Depending on the particulars of each agreement, any or all of the following may need to be specified. If they are relevant, they should be used throughout, modifying the standard form as appropriate.]
2. “Acceptance Criteria” means the written technical and operational performance and functional criteria and documentation standards set out in the [project plan.]
3. “Acceptance Date” means [the date for each Milestone when all Deliverables included in that Milestone have been accepted by Sponsor in accordance with the Acceptance Criteria and this Agreement.]
4. “Deliverable” means a deliverable specified in the [project plan.]
5. “Delivery Date” shall mean, [with respect to a particular Milestone,] the date on which University has delivered to Sponsor all of the Deliverables [for that Milestone] in accordance with [the project plan and] this Agreement.

6. “Documentation” means the documents, manuals and written materials (including end-user manuals) referenced, indicated or described in [the project plan] or otherwise developed pursuant to this Agreement.
7. “Milestone” means the completion and delivery of all of the Deliverables or other events which are included or described in [the project plan] scheduled for delivery and/or completion on a given target date; a Milestone will not be considered completed until the Acceptance Date has occurred with respect to all of the Deliverables for that Milestone.

4 DEVELOPMENT OF SOFTWARE

1. Senior Design Team will use its best efforts to develop the Software described in [the project plan.] The Software development will be under the direction of or his/her successors as mutually agreed to by the parties (“Team Lead”) and will be conducted by the Team Lead. The Team will deliver the Software to the satisfaction of the course instructor that reasonable effort has been made to address the needs of the client. The Team understands that failure to deliver the Software is grounds for failing the course.
2. Sponsor understands that the Senior Design course’s mission is education and advancement of knowledge, and, consequently, the development of Software must further that mission. The Senior Design Course does not guarantee specific results or any results, and the Software will be developed only on a best efforts basis. The Software is considered PROOF OF CONCEPT only and is NOT intended for commercial, medical, mission critical or industrial applications.
3. The Senior Design instructor will act as mediator between Sponsor and Team; and resolve any conflicts that may arise.

5 COMPENSATION

[This is entirely subject to negotiation. Normally NO COMPENSATION occurs in a Senior Design Project. On occasion an intern status and wage is appropriate.]

6 CONSULTATION AND REPORTS

1. Sponsor’s designated representative for consultation and communications with the Team Lead shall be _____ or such other person as Sponsor may from time to time designate to the Team Lead (“Designated Representative”).
2. During the Term of the Agreement, Sponsor’s representatives may consult informally with course instructor regarding the project, both personally and by telephone. Access to work carried on in University facilities, if any, in the course of this Agreement shall be entirely under the control of University personnel but shall be made available on a reasonable basis.
3. The Team Lead will submit written progress reports. At the conclusion of this Agreement, the Team Lead shall submit a comprehensive final report in the form of the formal course documentation at the conclusion of the Senior Design II course.

7 CONFIDENTIAL INFORMATION

1. The parties may wish, from time to time, in connection with work contemplated under this Agreement, to disclose confidential information to each other (“Confidential Information”). Each party will use reasonable efforts to prevent the disclosure of any of the other party’s Confidential Information to third parties for

a period of three (3) years after the termination of this Agreement, provided that the recipient party's obligation shall not apply to information that:

- (a) is not disclosed in writing or reduced to writing and so marked with an appropriate confidentiality legend within thirty (30) days of disclosure;
 - (b) is already in the recipient party's possession at the time of disclosure thereof;
 - (c) is or later becomes part of the public domain through no fault of the recipient party;
 - (d) is received from a third party having no obligations of confidentiality to the disclosing party;
 - (e) is independently developed by the recipient party; or
 - (f) is required by law or regulation to be disclosed.
2. In the event that information is required to be disclosed pursuant to subsection (6), the party required to make disclosure shall notify the other to allow that party to assert whatever exclusions or exemptions may be available to it under such law or regulation.

8 INTELLECTUAL PROPERTY RIGHTS

[Negotiated on a case-by-case basis. This must address who owns the algorithms and who owns the source code. For example: All deliverables become property of the Sponsor. Roughly: If the idea originates with the sponsor, or if a sponsor pays you to develop an idea, then they have legitimate claim to the IP. If the idea originates from the University (through faculty or staff) then the University has legitimate claim. If the idea is yours (student) and you develop it without external compensation then you have legitimate claim.]

9 WARRANTIES

The Senior Design Team represents and warrants to Sponsor that:

- 1. the Software is the original work of the Senior Design Team in each and all aspects;
- 2. the Software and its use do not infringe any copyright or trade secret rights of any third party.

No agreements will be made beyond items (1) and (2).

10 INDEMNITY

- 1. Sponsor is responsible for claims and damages, losses or expenses held against the Sponsor. [Sponsor may have something to add here.]
- 2. Sponsor shall indemnify and hold harmless the Senior Design Team, its affiliated companies and the officers, agents, directors and employees of the same from any and all claims and damages, losses or expenses, including attorney's fees, caused by any negligent act of Sponsor or any of Sponsor's agents, employees, subcontractors, or suppliers.
- 3. NEITHER PARTY TO THIS AGREEMENT NOR THEIR AFFILIATED COMPANIES, NOR THE OFFICERS, AGENTS, STUDENTS AND EMPLOYEES OF ANY OF THE FOREGOING, SHALL BE LIABLE TO ANY OTHER PARTY HERETO IN ANY ACTION OR CLAIM FOR CONSEQUENTIAL OR SPECIAL DAMAGES, LOSS OF PROFITS, LOSS OF OPPORTUNITY, LOSS OF PRODUCT OR LOSS OF USE, WHETHER THE ACTION IN WHICH RECOVERY OF DAMAGES IS SOUGHT IS BASED ON CONTRACT TORT (INCLUDING SOLE, CONCURRENT OR OTHER NEGLIGENCE AND STRICT

LIABILITY), STATUTE OR OTHERWISE. TO THE EXTENT PERMITTED BY LAW, ANY STATUTORY REMEDIES WHICH ARE INCONSISTENT WITH THE PROVISIONS OF THESE TERMS ARE WAIVED.

11 INDEPENDENT CONTRACTOR

For the purposes of this Agreement and all services to be provided hereunder, the parties shall be, and shall be deemed to be, independent contractors and not agents or employees of the other party. Neither party shall have authority to make any statements, representations or commitments of any kind, or to take any action which shall be binding on the other party, except as may be expressly provided for herein or authorized in writing.

12 TERM AND TERMINATION

1. This Agreement shall commence on the Effective Date and extend until the end of classes of the second semester of Senior Design (CSC 467), unless sooner terminated in accordance with the provisions of this Section ("Term").
2. This Agreement may be terminated by the written agreement of both parties.
3. In the event that either party shall be in default of its materials obligations under this Agreement and shall fail to remedy such default within thirty (30) days after receipt of written notice thereof, this Agreement shall terminate upon expiration of the thirty (30) day period.
4. Any provisions of this Agreement which by their nature extend beyond termination shall survive such termination.

13 ATTACHMENTS

Attachments A and B are incorporated and made a part of this Agreement for all purposes.

14 GENERAL

1. This Agreement constitutes the entire and only agreement between the parties relating to the Senior Design Course, and all prior negotiations, representations, agreements and understandings are superseded hereby. No agreements altering or supplementing the terms hereof may be made except by means of a written document signed by the duly authorized representatives of the parties.
2. This Agreement shall be governed by, construed, and enforced in accordance with the internal laws of the State of South Dakota.

15 SIGNATURES

Replace with name of student #1

Date

Replace with name of student #2

Date

Replace with name of student #3

Date

Replace with name of sponsor's representative

Date

A

Product Description

Write a description of the product to be developed. Use sectioning commands as neccessary.

NOTE: *This is part of the contract.*

B

Publications

Research Track: This chapter will include any publications generated from the research. Most likely these will be preprints and one will just include the pdf.

C

Sprint Reports

1 Sprint Report #1

The sprint reports should be inserted here. Reports focus on process. Design elements can be inserted into the design chapter with the report discussing the design element in more of an overview fashion. If you write your sprint reports in \LaTeX it will be easier to include.

2 Sprint Report #2

3 Sprint Report #3

4 Sprint Report ...

D

Industrial Experience and Resumes

1 Resumes

Your resumes are included here. See the source file (industrial.tex) and uncomment the PDF includes to see how this works. If your resume is written in \LaTeX then you can just insert the \LaTeX source code.

2 ABET: Industrial Experience Reports

2.1 Name1

2.2 Name2

2.3 Name3

E

Acknowledgment

Thanks

F

Supporting Materials

This document will contain several appendices used as a way to separate out major component details, logic details, or tables of information. Use of this structure will help keep the document clean, readable, and organized.

L^AT_EX Example

L^AT_EX sample file: [Remove from submitted materials](#)

1 Introduction

This is a sample input file. Comparing it with the output it generates can show you how to produce a simple document of your own.

2 Ordinary Text

The ends of words and sentences are marked by spaces. It doesn't matter how many spaces you type; one is as good as 100. The end of a line counts as a space.

One or more blank lines denote the end of a paragraph.

Since any number of consecutive spaces are treated like a single one, the formatting of the input file makes no difference to T_EX, but it makes a difference to you. When you use L^AT_EX, making your input file as easy to read as possible will be a great help as you write your document and when you change it. This sample file shows how you can add comments to your own input file.

Because printing is different from typewriting, there are a number of things that you have to do differently when preparing an input file than if you were just typing the document directly. Quotation marks like “this” have to be handled specially, as do quotes within quotes: “ ‘this’ is what I just wrote, not ‘that’ ”.

Dashes come in three sizes: an intra-word dash, a medium dash for number ranges like 1–2, and a punctuation dash—like this.

A sentence-ending space should be larger than the space between words within a sentence. You sometimes have to type special commands in conjunction with punctuation characters to get this right, as in the following sentence. Gnats, gnus, etc. all begin with G. You should check the spaces after periods when reading your output to make sure you haven't forgotten any special cases. Generating an ellipsis . . . with the right spacing around the periods requires a special command.

T_EX interprets some common characters as commands, so you must type special commands to generate them. These characters include the following: \$ & % # { and }.

In printing, text is emphasized by using an *italic* type style.

A long segment of text can also be emphasized in this way. Text within such a segment given additional emphasis with Roman type. Italic type loses its ability to emphasize and become simply distracting when used excessively.

It is sometimes necessary to prevent T_EX from breaking a line where it might otherwise do so. This may be at a space, as between the “Mr.” and “Jones” in “Mr. Jones”, or within a word—especially when the word is a symbol like *itemnum* that makes little sense when hyphenated across lines.

Footnotes¹ pose no problem.

T_EX is good at typesetting mathematical formulas like $x - 3y = 7$ or $a_1 > x^{2n}/y^{2n} > x'$. Remember that a letter like x is a formula when it denotes a mathematical symbol, and should be treated as one.

¹This is an example of a footnote.

3 Displayed Text

Text is displayed by indenting it from the left margin. Quotations are commonly displayed. There are short quotations

This is a short a quotation. It consists of a single paragraph of text. There is no paragraph indentation. and longer ones.

This is a longer quotation. It consists of two paragraphs of text. The beginning of each paragraph is indicated by an extra indentation.

This is the second paragraph of the quotation. It is just as dull as the first paragraph.

Another frequently-displayed structure is a list. The following is an example of an *itemized* list.

- This is the first item of an itemized list. Each item in the list is marked with a “tick”. The document style determines what kind of tick mark is used.
- This is the second item of the list. It contains another list nested inside it. The inner list is an *enumerated* list.
 1. This is the first item of an enumerated list that is nested within the itemized list.
 2. This is the second item of the inner list. \LaTeX allows you to nest lists deeper than you really should.

This is the rest of the second item of the outer list. It is no more interesting than any other part of the item.

- This is the third item of the list.

You can even display poetry.

There is an environment for verse
Whose features some poets will curse.

For instead of making
Them do *all* line breaking,
It allows them to put too many words on a line when they'd rather be forced to be terse.

Mathematical formulas may also be displayed. A displayed formula is one-line long; multi-line formulas require special formatting instructions.

$$x' + y^2 = z_i^2$$

Don't start a paragraph with a displayed equation, nor make one a paragraph by itself.

4 Build process

To build \LaTeX documents you need the latex program. It is free and available on all operating systems. Download and install. Many of us use the TexLive distribution and are very happy with it. You can use a editor and command line or use an IDE. To build this document via command line:

```
alta> pdflatex SystemTemplate
```

If you change the bib entries, then you need to update the bib files:

```
alta> pdflatex SystemTemplate
```

```
alta> bibtex SystemTemplate
```

```
alta> pdflatex SystemTemplate
```

```
alta> pdflatex SystemTemplate
```

The template files provided also contain a Makefile, which will make things much easier.

Acknowledgment

Thanks to Leslie Lamport.