

MICHIGAN STATE UNIVERSITY  
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING  
ECE 480 SENIOR DESIGN

---

**Final Project Proposal**  
**ArcelorMittal**  
**Safety Equipment Bar Code Scanner**

---

*Team Members:*

Kyle INCH  
Alexandria MARONE  
Seth MCKISSON  
Trevor SABO  
Ian GROSH

**Design Team**  
**Three**

*Facilitator:*

Dr. Bingsen WANG

October 14, 2016



**Executive Summary**

The purpose of this project is to create a way to electronically keep track of safety inspection information for various safety equipment in ArcelorMittal plants. This will improve the current method of manual, paper records and minimize data loss. This will also allow inspection results to be automatically consolidated in one central database. Reports will then be generated from the central database to determine items in each location that pass or fail inspection, drastically improving the current record-keeping system of safety equipment at ArcelorMittal.

# Contents

<b>I</b>	<b>Introduction</b>	<b>3</b>
<b>II</b>	<b>Overview of Customer's Requirements</b>	<b>3</b>
<b>1</b>	<b>Background Research</b>	<b>3</b>
1.1	Assessing Customer Needs . . . . .	3
1.2	Design Constraints . . . . .	3
1.3	Feasibility of Designs . . . . .	3
1.4	Risks . . . . .	4
1.5	Overview of System Models . . . . .	4
1.6	Developmental Strategy . . . . .	4
<b>2</b>	<b>High Level Languages</b>	<b>4</b>
2.1	Languages . . . . .	4
<b>III</b>	<b>Technical</b>	<b>5</b>
<b>3</b>	<b>Project Definition</b>	<b>5</b>
3.1	Scope . . . . .	5
3.2	Function Definition . . . . .	6
3.3	Use example . . . . .	7
<b>4</b>	<b>Technical Design</b>	<b>7</b>
4.1	Mobile Application . . . . .	7
4.1.1	Mobile User Interface & Experience . . . . .	8
4.1.2	Barcode Scanning . . . . .	8
4.1.3	Off-line Mode . . . . .	8
4.2	Web Application . . . . .	9
4.3	Database . . . . .	10
<b>IV</b>	<b>Cost</b>	<b>10</b>
<b>V</b>	<b>Project Management</b>	<b>12</b>
<b>5</b>	<b>Scheduling</b>	<b>12</b>
5.1	Preliminary Set Up . . . . .	12
5.2	Interconnectivity . . . . .	13
5.3	Preliminary Application Development . . . . .	13
5.4	Secondary Application Development . . . . .	14

## List of Figures

1	System Components . . . . .	5
2	FAST Digram . . . . .	6
3	Mock up of Start Screen . . . . .	8
4	Mock up of splash screen . . . . .	8
5	Database entity relations Diagram . . . . .	11
6	Gantt Chart . . . . .	15

## List of Tables

1	Web Framework Solution Selection Matrix . . . . .	10
2	Database Solution Selection Matrix . . . . .	11

## **Part I**

# **Introduction**

Design team three has been asked to create a system to keep track of safety equipment on ArcelorMittal's buildings. To do this the team needs to build some systems. These systems will enable administrators to both monitor compliance standards on areas they are in charge of, and make sure that safety equipment is being properly checked and documented. Reports will be sent out periodically to administrators. On the user end, an Android application that uses a camera to scan barcodes will be created. This enables users to quickly answer questions regarding safety equipment standards.

## **Part II**

# **Overview of Customer's Requirements**

## **1 Background Research**

There is background research that needs to be implemented to characterize primary hardware and software components that are needed for a successful project.

### **1.1 Assessing Customer Needs**

The customer's needs, as well as the needs of their users, are team three's highest priority. Team three's sponsor, primary point of contact, is Jim Lang with ArcelorMittal. Team three's sponsor clearly defines ArcelorMittal's needs, and how the user will be interacting with the system. The customer's needs are clearly documented by ArcelorMittal. This documentation assesses the needs and interests of the ultimate user.

### **1.2 Design Constraints**

With customer needs comes design constraints. The first design constraint is time. Team three has been given one semester to deliver a complete solution. Another constraint is cost. The team has been given a budget of \$500 to create this system.

### **1.3 Feasibility of Designs**

Criteria to judge the feasibility of the design has been outlined by ArcelorMittal. Team three was given requirements necessary for designing and implementing a minimum viable product. The requirements are based on ArcelorMittal's current implementation of problem, Field ID. To satisfy necessary criteria, team three's goal is to implement a software-based solution; hence, automating the entire process. Ultimately feasibility will be judged based on universal usability. In order to rank the feasible designs, team three must also consider criteria based on speed of response time, organization of documentation, report generation, and ability to publish software development environment into production.

Each set of conceptual designs/variations meets the feasibility criteria. Team three's solution

meets the goals of the customer's needs; the proposed solution has been approved by both team sponsor and facilitator. Trade off studies that need be conducted to assess the above concerns are speed of response time, and difficulty to move team three's software development environment to production. In a software environment, feasible conceptual designs are always subjected to change.

## **1.4 Risks**

A majority of risks revolve around the security of team three's proposed system. This is solved by the implementation and integration on ArcelorMittal's servers. ArcelorMittal has their own security team to ensure that the proposed system once moved to a production development will be secure. In addition the data team three is dealing with is not confidential. That being said, all code will be written in a secure manner. Special attention will be paid to SQL queries, and injections. Due to safety and liability restrictions, team three is unable to gain access to ArcelorMittal's systems; most specifically their databases. Team three's solution is designed to be integrated into ArcelorMittal's existing system.

## **1.5 Overview of System Models**

As the nature of this project is more software oriented then hardware, developing models for system sensors, actuators, interfaces or other principle level system components are not a concern. As a result, no validation is needed. Similarly, issues involving control processes are not a concern. However, Operating Systems at different interconnections distances is a concern. ArcelorMittal has a special requirement that team three's mobile application must operate without WIFI; this issue must be accommodated. Additionally storage on the Nexus 7 tablet is also a concern. A limit of 16GB of memory is crucial; therefore, team three's application must be well within that limit. Finally, the software must be compatible with hardware specifics of the Nexus 7, which are a ULP GeForce graphics card, Quad Core processors, and 1GB of RAM [4].

## **1.6 Developmental Strategy**

Team three has picked a bottom-up approach. The team will implement lower levels specifics at highest level of importance. This gives the team the ability to test database communication first. Further, when fusing data into the user interface, the API implemented creates connections for testing. This allows the team to efficiently design an experience while minimizing on constraints. Developmental tools for the environments will also be utilized for testing such as Android Studio for mobile development.

# **2 High Level Languages**

## **2.1 Languages**

As for compilers on the back-end most things will be written in the programming language Python, which does not have a compiler. For the email service, SMTP Python libraries will be used, as well as various web framework libraries. The web framework libraries are provided by Django, a high-level Python Web framework.

Android Application Development is written in Java which does have a compiler. On the administrator side, jQuery and Twitter's Bootstrap will be used for web development. A detailed implementation can be found in section 4.2.

## Part III

# Technical

### 3 Project Definition

This team has been assigned an industry sponsored project from ArcelorMittal, who need a way to track their industrial safety equipment within their buildings. In order to build the specified system team three will need to build three primary systems. The first being an Android application for operators to scan barcodes on safety locations and equipment. Second, a web application to allow administrators to specify questions for specific pieces of safety equipment, and furthermore assign responsibility to workers. Finally, a server infrastructure will be created to ensure data is held properly and securely, host the web application, connect to the Android application, and generate reports. Below is the layout.

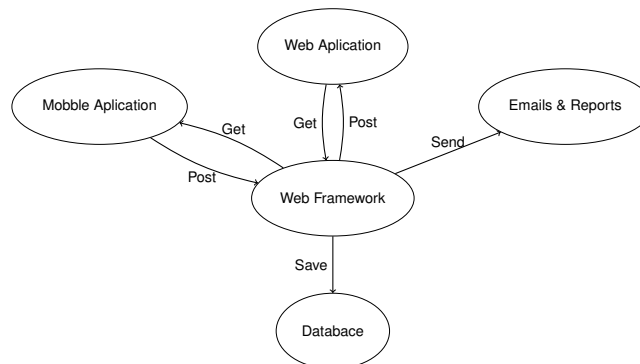


Figure 1: System Components

#### 3.1 Scope

In order to guide the team's design project and ensure the project has limits of what the team will build, a project scope has been defined:

- **Mobile Application**

- Off-line mode
- Barcode scanning
- Barcode based questions

- **Sever Side Database and Middle-ware**

- Host:  
Web application  
Database API for mobile applica-  
tion

- **Emails & Reports**

- Inform administrators of:  
Failing devices  
Delinquent inspectors
- Inform inspectors of:  
Upcoming inspections  
Missed inspections

- **Web Application**

- Add locations to the database
- Add safety equipment types to the database
- Create questions
- Associate barcodes with:  
Locations  
Safety equipment
- Associate locations with:  
Safety equipment  
Questions
- Associate safety equipment with  
Locations  
Questions
- Add questions to reports
- Create timetable for reports
- Add recipient to reports

### 3.2 Function Definition

In order to justify the existence of items in Part 3.1 the team created a number of function definitions. These definitions were then consolidated into the FAST diagram in Figure 2 along with a more detailed description. The Primary function for the project is **Ensure Compliance** This is

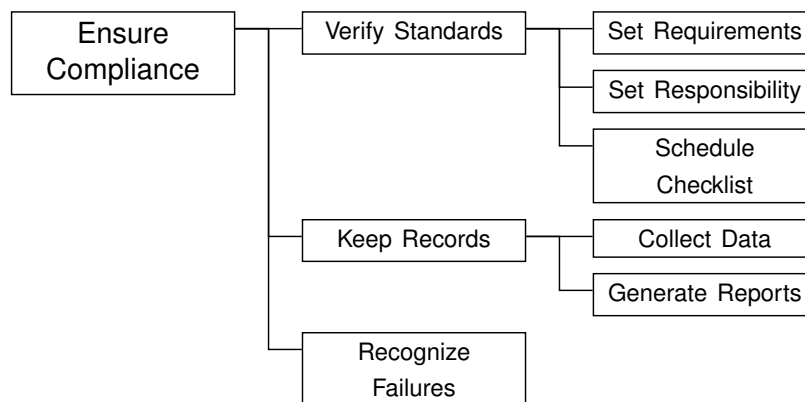


Figure 2: FAST Digram

the main goal of the system the team has been commissioned to build. From the team's primary function are derived two secondary functions each having there own tertiary functions:

- **Verify Standards:**

In order to ensure compliance with all pertinent safety regulations the system must be able to check the and verify that all the standards are being upheld in the various locations and across the numerous safety devices.

- **Keep Records:**  
In order to ensure that all of the safety laws and regulations are followed, the team must keep records of all the locations which must have safety equipment present in a building, what safety equipment must be present, and how to verify that it is in working condition.
- **Recognize Failures:**  
In order to recognize failures, the team needs to recognize when a piece of safety equipment is not in compliance. This could mean a variety of things, the equipment could not pass the safety inspections, or the equipment could not be present.

From the secondary function Keep Records the team has derived:

- **Generate Reports:**  
In order to keep records and ensure that every location and item is in compliance the system must be able to generate reports on any set of data.

From the secondary function Verify Standards the team has derived:

- **Set Requirements:**  
In order to verify standards the system must be able to set compliance requirements for each location and item which that is tracked by the system.
- **Set Responsibility:**  
A person must be assigned responsibility for a number of locations and items within the system. Once this Responsibility is set the owner can be held accountable for their set of locations and items.
- **Schedule Checklists:**  
Checklists will be created for users to show what work is left outstanding.
- **Collect Data:**  
Data will be collected based on user's answers to inspection questions.

### **3.3 Use example**

Here the team will use the project description to provide an example of how the system will be used.

## **4 Technical Design**

### **4.1 Mobile Application**

Design team three will build a mobile application for a tablet. This will allow a user to download a local copy of the database. The database holds safety equipment inspection questions. The user can then operate without access to the internet and scan barcodes of safety devices to answer a series of inspection questions. The questions' answers are pushed back to the database over the sponsor's internet. This last data push will be completed when the user gets back to their desk, where there is WIFI available.



### 4.1.1 Mobile User Interface & Experience

Design team three will create a user interface and experience (UI/UX) to guide the safety inspector through the process of an inspection. In Figure 3, the user is prompted to login so that the data relevant to that user can be synced to the mobile device. By doing so that the inspection results can be recorded under the correct employee's name.

If the user's device is connected to the internet, they will press the button labeled Sync in Figure 4 which will retrieve the inspection locations and safety device information pertinent to the user. Once the inspector reaches a location they will press the Scan Barcode Location button shown in Figure 4.



Figure 3: Mock up of Start Screen

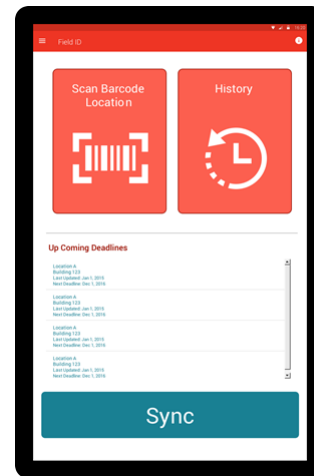


Figure 4: Mock up of splash screen

### 4.1.2 Barcode Scanning

The user will scan the barcode with the Android application created. After successful scan, questions will pop up that need to be answered. On the event of a non-successful scan, a manual barcode insertion will be needed. The barcode scanning is done with a ZXing library (Zebra Crossing), and is open source and very well documented. Through testing team three has found that this library produces a very reliable, accurate scanner and is perfect for the design specifications.

In comparison, the Google Barcode image recognition software was not quite what the team wanted. It is not open source, but it was free. Team three choose the ZXing library because it was easier to use.

### 4.1.3 Off-line Mode

The Android application will be able to run without WIFI. This is very important because at the time of the scan, there will be no WIFI. Data collected from inspections will be stored locally on the tablet until WIFI is available for sync.

## 4.2 Web Application

Administrators need to be able to add safety equipment and inspection questions to the database, assign ownership, and generate reports. Design team three must build a web application to satisfy that requirement. In order to do so, the design team will use a web application development framework to build the features required quickly and robustly.

Based on the technical requirements and anticipated need team three has devised the following criteria to judge possible choices of web frameworks:

- **Integrates With Sponsor IT**  
Design team three's sponsor ArcelorMittal, uses a Microsoft Windows based software for their internal IT. The team must ensure their choice of framework is compatible in a Microsoft server environment.
- **Documentation**  
One of the most important parts of any software library is the documentation. If a developer is required to dig through source code to determine how a library works, the amount of time which will be required will increase exponentially and will result in code that is more prone to defects.
- **Ease Of Use**  
The complexity of a framework can be quite large which can partially be determined by the framework's programming language. This requires design team three to determine how difficult standard tasks such as serving a web page, and dynamically creating content will be with each framework.
- **Learning Curve**  
The learning curve is going to deal with how much the team knows about said framework, and how fast other people have learned it. To assess the learning curve, the team must take advantage of software libraries and documentation. If it seems that the tutorial is very complex, that framework is probably not the best for the team.
- **Stack Fullness**  
Stack fullness refers to how well each part of the stack is covered in the proposed framework. Since the team is creating a Full Stack application (an application that has a front-end and a back-end), this is going to be very important to consider.
- **Third Party Ecosystem**  
The third party ecosystem deals with the integration from the team's sponsor, ArcelorMittal. Team three needs to make sure that the product that is made has an easy integration with ArcelorMittal's systems.

Next, the team must choose from the plethora of existing frameworks. Seven of which are commonly used in web application development to compare against these requirements. Each criteria is marked on a log scale of either 1,3, or 9 and displayed in Table 1.

- **Ruby on Rails**  
Ruby on Rails is a web framework also known as Rails built in the programming language Ruby. It is open source and free. Rails supports web standards such as JSON, or XML for data transfer. Ruby on Rails is a great candidate for the framework team three will use, but team three has more experience with Python than Ruby [7].
- **Django**  
Django is a Python built framework and is one the most popular web frameworks. Django has numerous built in interfaces and a healthy ecosystem of free and open third-party addons. Django has a very easy learning curve, because it is built in Python. The framework is fast, scalable and secure [1].
- **Play**

Play is a Java Web framework. Play is based on a lightweight, stateless, web-friendly architecture. It provides predictable and minimal resource consumption (CPU, memory, threads) for highly-scalable applications. This framework is beneficial to team three's solution as their mobile application is also written in Java[5].

- **Express.js**

Express.js is a minimal and flexible Node.js web application framework that provides a robust set of features for web and mobile applications, as well as APIs. It is relatively minimal with many features available as plugins. [2].

- **Laravel**

Laravel is a PHP web framework that utilizes the model-view-control architectural pattern. Laravel has an efficient dependency manager and provides built-in ways to access relational databases. It also contains utilities for application deployment and maintenance. [3].

- **Revel**

Revel is the Go language web framework. None of the engineers on team three have Go experience. Revel is relatively new so there is not much documentation for trouble shooting. Revel compiles code in templates for the developer automatically. It also gives descriptive error messages, making it easier to debug. However we are looking for something with more familiarity. [6].

- **TurboGears**

This is a next generation Python web framework. Taking cues from frameworks like Django and Rails, this framework uses advanced Python and other languages to build a web application. This is used for more complex applications, and is not very well supported [8].

	Django	Rails	Play	Laravel	Revel	Flask	TurboGears
Integrates With Sponsor IT	3	3	3	3	1	1	1
Documentation	9	3	3	3	9	3	3
Ease Of use	9	1	3	3	1	9	9
Learning curve	3	1	1	1	1	3	1
Stack fullness	9	9	9	9	9	1	9
Third party Ecosystem	9	9	9	9	3	1	3
<b>Total</b>	<b>42</b>	<b>26</b>	<b>28</b>	<b>28</b>	<b>24</b>	<b>18</b>	<b>26</b>

Table 1: Web Framework Solution Selection Matrix

### 4.3 Database

In order to see how items in the system are related to each other the team has created a entity relation diagram.

	SQLite	Microsoft SQL	Oracle Database	IBM DB2	SAP SQL	MySQL	PostgreSQL	Firebird SQL	MongoDB	Hadoop
Integrates with Sponsor IT	3	9	9	9	3	9	3	3	1	1
Ease of Use	9	1	1	1	1	3	3	3	1	1
Documentation	9	3	3	1	1	3	1	1	3	1
Django Support	9	3	9	3	3	9	9	3	1	1
Fast Insertion	1	3	3	3	3	3	3	3	9	9
Large Data Sets	1	9	9	9	9	3	3	3	9	9
Enterprise Security	1	9	9	9	9	3	3	3	3	3
Team Experience	9	1	1	1	1	3	1	1	3	1
The Correct Price	9	9	1	1	1	9	9	9	9	9
<b>Total</b>	<b>51</b>	<b>47</b>	<b>45</b>	<b>37</b>	<b>31</b>	<b>45</b>	<b>35</b>	<b>29</b>	<b>39</b>	<b>35</b>

Table 2: Database Solution Selection Matrix

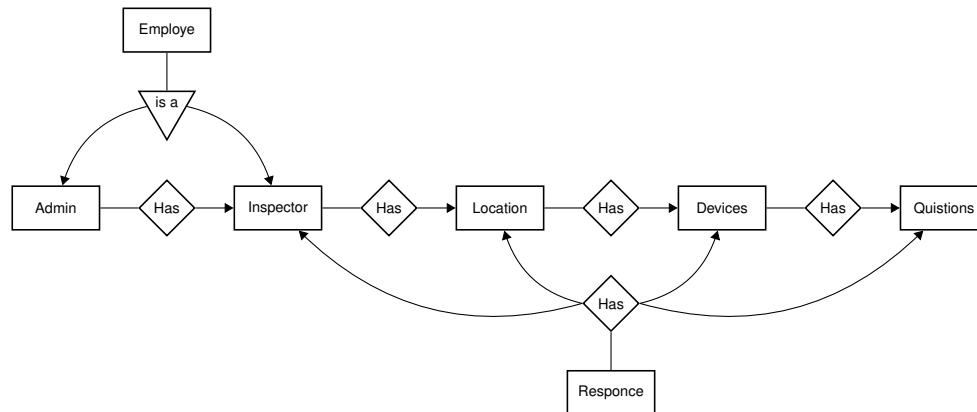


Figure 5: Database entity relations Diagram

## Part IV

# Cost

The only cost incurred by the design team will be purchasing a Nexus 7 tablet. The tablet will be of the same model and year or similar to the sponsor's devices. The Nexus 7 tablet's cost averages to be about \$180. The Nexus 7 tablet is a crucial part to this design because the team needs a device to test our software on. There are Android development studios for developing, but the team needs to make sure that the software will work for production. Due to ArcelorMittal's security measures, we are unable to use one of the sponsor's own devices. To speed up the process of development, a group member has volunteered their Nexus 7 tablet to be used.

- **Tablet**

The Nexus 7 Tablet \$179.90 [4].

## Part V

# Project Management

## 5 Scheduling

In order to better adapt to the complex requirements defined in Section 3, the team has broken the tasks into four iterative cycles. Each cycle is planned to address a number of the design requirements, which must be developed at the same time, due to the highly interconnected nature of the individual subsystems.

### 5.1 Preliminary Set Up

Several instances of contact with the sponsor were made for preliminary set up in order to both better understand the customer's needs and build a relationship for ongoing communication. At the end of this cycle the team will deliver a project proposal to the faculty advisor and the project sponsor along with verifying the basic functionality of the database, web application, Android application and design mock-ups.

- **Meet with Sponsor:** (The Team)  
The team prepares questions and meets in person with the ArcelorMittal sponsor in order to better understand the needs defined in the project description. In this meeting the project sponsor was asked to describe key features and furthermore, what a successful project looked like.
- **Solidify Understanding of Project:** (The Team)  
In this task, the team meets to discuss what was learned in the meeting with the project sponsor previously. This assisted the team to build the requested system, because notes were taken immediately after the meeting to ensure nothing was left out.
- **Additional Questions for Sponsor:** (The Team)  
After discussion on the both the high level work flows and technical feasibility of the project, the team will compose a set of questions to be electronically mailed to the project sponsor in order to clear up lingering discontinuity in the teams understanding of the system.
- **Set up Database:** (Alexandria & Ian)  
In this task, two team members will decide on a framework for building the server side infrastructure for the system. This will include choosing and setting up the server operating system, choosing the main programming language to be used for building the sever infrastructure.
- **Proof of Concept for Barcode Scanning:** (Kyle)  
In this task a team member will build a simple program on an Android tablet. This initial design will be able to scan a barcode and read the ID printed on the barcode. This task will also involve exploring several different scanning algorithms and assessing which is most effecient.
- **Draw Mock ups:** (Trevor)  
In this task a team member will draw the initial designs for the user experience that will be used. This gives the project sponsor an idea of what the user interface will look like.

- **Set up Web Page:** (Seth)

In this task a team member will setup an initial front-end for the administrator website. The team member will decide on a set of predefined web objects that can be used to build web applications.

## 5.2 Interconnectivity

In the interconnectivity step, design team three will focus on building and testing connections between the baseline system, setup in the section 5.1.

- **Web speaks to Database:** (Seth)

This step will consist of establishing a connection between the existing front-end of the administrator website developed in section 3.1 and the database that will allow the administrator site to read from and write to the database. This functionality will be used for retrieving and modifying information regarding the whereabouts and inspection status of safety equipment.

- **Mobile speaks to Database:** (Kyle)

This development task will require establishing a connection between the Android mobile application and the database. This will allow the mobile application to read information from the database to be displayed in the mobile application. The application can then upload new inspection results from a local instance to the central database.

- **Demonstrate to Sponsor:** (The Team)

In this phase, team three will demonstrate to the sponsor, either in person or via video, the ability of the website and mobile application to both read and post information to one common, central database to be used for storing inspection results and safety equipment details.

- **Mock ups and Wire Frames:** (Trevor)

The user interface of the mobile application and the administrator website will be modified after receiving sponsor feedback regarding their current design to better suit the needs and wants of the customer.

## 5.3 Preliminary Application Development

- **Mobile Application Development:** (Kyle)

The implementation of the mobile application will be completed during this phase to produce a fully functioning prototype that is able to scan a barcode and pull information from the database related to that barcode. The application will then be able to upload inspection results of each item over a WIFI network to the central database.

- **Web Application Development:** (Seth and Alexandria)

The front-end and back-end of the administrator website will be completed during this phase to produce an initial prototype. The administrator website will be able to receive input from a user regarding barcodes and their related specifications and post that information to the database to be used by the mobile application. This development is to be done in parallel with the mobile application.

- **Database Refinements:** (Ian and Alexandria)  
The database will be continually refined throughout the development of the administrator website and mobile application to address any issues discovered during this development phase.
- **Feedback:** (The Team)  
The team will demonstrate the functioning prototype of both the mobile application and administrator website to the sponsor and show how they integrate together to achieve a solution to the original problem. The team will discuss the sponsor's opinion regarding pros and cons of the implementation and create a plan for any modifications that the customer desires.
- **New Mock ups:** (Trevor)  
Modification to the user interface design will be made if the sponsor desires.

## 5.4 Secondary Application Development

- **Mobile Application Development:** (Kyle)  
Any desired design modifications to the mobile application requested by the sponsor in section 1.1 will be implemented and tested in this phase.
- **Web Application Development:** (Seth and Alexandria)  
Any desired design modifications to the administrator website requested by the sponsor in section 1.1 will be implemented and tested in this phase.
- **Database Refinements:** (Alexandria)  
Any necessary database changes needed to accommodate modifications made to the mobile application or administrator website will be implemented.
- **Make project shippable to sponsor:** (The Team)  
The mobile application will be compiled into an executable that can be uploaded on the sponsor's tablets. All passwords necessary to access the administrator website will be given to the sponsor along with the source code for the website so that the website may be uploaded and hosted wherever the sponsor desires.

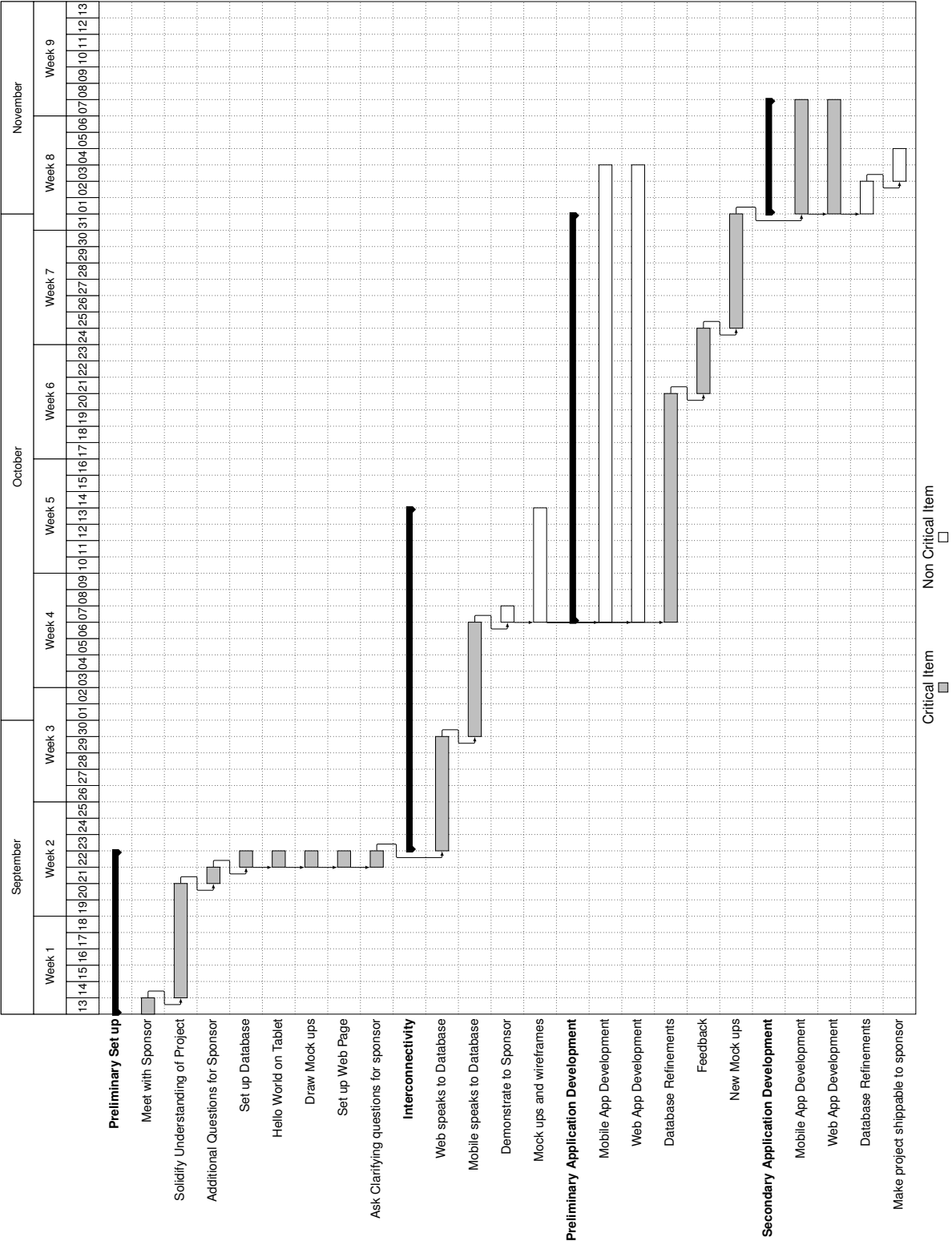


Figure 6: Gantt Chart



## References

- [1] Django Documentation. <https://docs.djangoproject.com/en/1.10/>, 2016.
- [2] Express. <http://expressjs.com/>, 2016.
- [3] Laravel. <https://laravel.com/docs/5.3>, 2016.
- [4] Google™ Nexus 7 Tablet 16gb. [http://www.staples.com/Google-Nexus-7-Tablet-16GB/product\\_215186](http://www.staples.com/Google-Nexus-7-Tablet-16GB/product_215186), 2015.
- [5] Play Documentation. <https://www.playframework.com/documentation/2.5.x/Home>, 2016.
- [6] Revel. <https://revel.github.io/manual/concepts.html>, 2016.
- [7] Welcome to Rails. <http://api.rubyonrails.org/>, 2016.
- [8] Turbogears Features. <http://turbogears.org/features.html>, 2016.