SANTA CLARA UNIVERSITY
DEPARTMENT OF COMPUTER ENGINEERING

Date: December 2, 2016

I HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER MY SUPERVISION BY

Karsten Andersenl
Matt Gibson

ENTITLED

On the Relative Positioning of Mobile Devices Through Bluetooth Analysis

BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

BACHELOR OF SCIENCE IN COMPUTER SCIENCE AND ENGINEERING

_____
Thesis Advisor

_____
Thesis Advisor

_____
Department Chair

_____
Department Chair

On the Relative Positioning of Mobile Devices Through Bluetooth Analysis

by

Karsten Andersenl
Matt Gibson

Submitted in partial fulfillment of the requirements
for the degree of
Bachelor of Science in Computer Science and Engineering
School of Engineering
Santa Clara University

Santa Clara, California
December 2, 2016

On the Relative Positioning of Mobile Devices Through Bluetooth Analysis

Karsten Andersenl
Matt Gibson

Department of Computer Engineering
Santa Clara University
December 2, 2016

ABSTRACT

Currently, there is no way to accurately track a mobile devices position in a given space without the use of GPS or WiFi. This means that people who are interested in understanding where they currently are in relationship to other people or landmarks and do not have access to these technologies are not able to get accurate location information. In many cases, GPS and WiFi fail to provide the accuracy required to provide useful information as well.

The solution being proposed is a software layer that utilizes existing bluetooth hardware to create a highly accurate relative positioning system. The software layer will utilize a variety of algorithms to interpret the relative signal strength of known bluetooth transmitters and will be able to return a useful, three dimensional, representation of where a smartphone or other bluetooth device is. The proposed framework offers a software solution that will be cheap and highly distributable due to the fact that is utilizes existing technologies.

Table of Contents

List of Figures

Chapter 1

Introduction

1.1 Motivation

Currently, there is no way to accurately track a mobile devices position in a given space without the use of GPS or WiFi. This means that people who are interested in understanding where they currently are in relationship to other people or landmarks and do not have access to these technologies are not able to get accurate location information. While most people with modern smartphones do have nearly unlimited access to GPS and WiFi, this does not necessarily exclude them from the group of people who are affected by this issue. In many cases, GPS and WiFi fail to provide the fine grain accuracy required to provide useful information. Within a building GPS and WiFi can be unreliable because their physical signature is fundamentally not designed to address the need for indoor or finely tuned positioning. This can cause the distance and elevation that GPS reports back to lose accuracy. It cannot determine what floor youre on so if youre on the second floor or the ninth floor, you would show up in the same location. In terms of distance, it could show you at least several meters off from your real world location. Best case scenario, GPS will get someone to the door of a building they are looking for, but after that the person is on their own once inside of that building. Due to their design, GPS and WiFi cannot help you determine your location once inside a building in many cases, especially if the building has multiple floors. Often times stores and campuses provide physical maps for visitors. For a physical map however, there is no map for finding the map. To use the map, a person physically has to move to it to view where they are. Also, if there is a single map for all visitors, then a person cant carry it with them once they head for their destination. This means people can get lost on the way to their destination after leaving the map. Other current solutions try to solve this issue with proximity detection technology. The problem here is that proximity based systems can only tell a person how close they are relative to something; it doesnt give information on the direction or elevation of whatever the person may be

looking for. This can cause people to waste time figuring out what direction they need to head in or even put them a floor above or below their destination.

1.2 Solution

The solution being proposed is a software layer that utilizes existing bluetooth hardware to create a highly accurate relative positioning system. The software layer will utilize a variety of algorithms to interpret the relative signal strength of known bluetooth transmitters and will be able to return a useful, three dimensional, representation of where a smartphone or other bluetooth device is. The ability to understand the location of a device in two or three dimensions as opposed to the proximity of a device in one dimension allows for an exponential increase in the possible applications for such a positioning framework. Additionally, the ability to navigate using a smartphone indoors vastly improves the currently nondigital solution of central physical maps that are currently in use in many large buildings. The proposed framework offers a software solution that will be cheap and highly distributable due to the fact that is utilizes existing technologies. This means the framework will be able to support a wide variety of applications without needing to wait around for the next breakthrough spectrum standard in positioning technology.

Chapter 2

Requirements

2.1 Functional Requirements

Critical:

- The system will return a location when 3+ beacons are available
- The system will be able to interpret BLE beacon noise levels to produce a precise location
- The system will give more accurate location data than GPS or WiFi in specific situations

Recommended:

- Offline installation of system does not undermine its usability for devs and admins

2.2 Non-Functional Requirements

Critical:

- The framework is easy to use for developers
- The framework will return location coordinates with an security between 0m and 5m
- The system will scale to large applications

Recommended:

- Admin interface is easy to use
- Framework is optimized for battery conservations

2.3 Design Constraints

- System must work on iOS
- Must use low cost BLE beacons
- System conforms to iBeacon protocol

Chapter 3

Use Cases

Displays the use case diagram and the description of each. The purpose is to define tasks an actor will perform to achieve a certain goal.

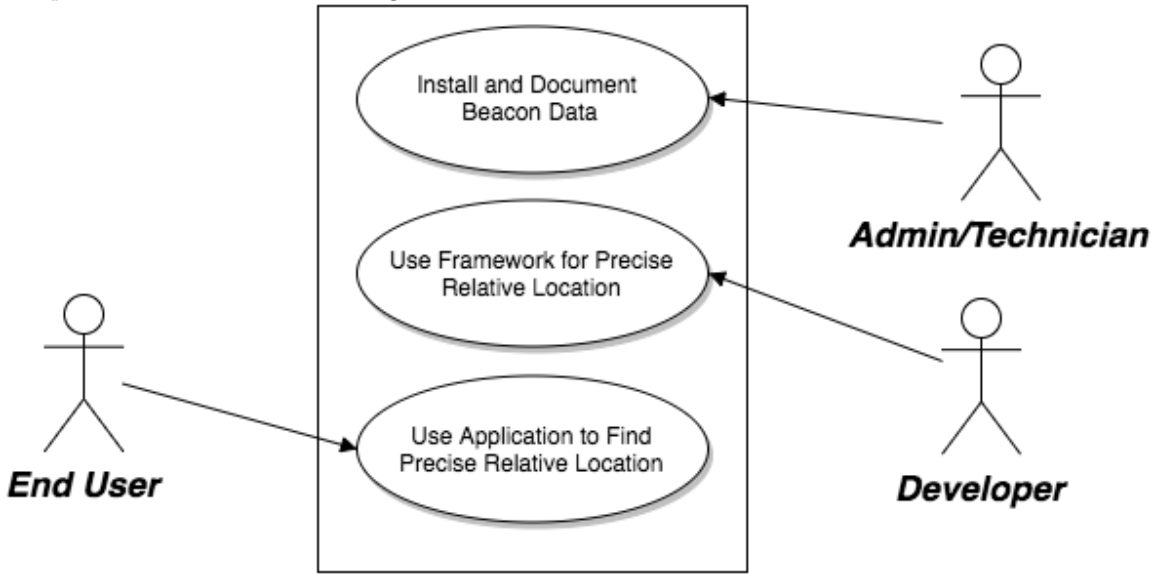

Figure 3.1: Use Case Diagram

1.

- Name: Install and Document Beacon Data
- Goal: To install and load location and other info about bluetooth beacons
- Actors: Admin/Technician
- Preconditions:Bluetooth beacons must be placed and initialized
- Steps: Location and other information is stored into a database
- Post-conditions: Database will have bluetooth beacon data loaded
- Exceptions: Information uploaded is incorrect

2.

- Name: Use Framework for Precise Relative Location
- Goal: To use framework to help application to discover the device's precise relative location
- Actors: Developer
- Preconditions:Framework must be loaded
- Steps: Developer writes code to utilize framework
- Post-conditions: Application will be able to discover the device's precise relative location
- Exceptions: Framework isn't loaded or used properly

3.

- Name: Use Application to Find Precise Relative Location
- Goal: To use application to discover the device's precise relative location
- Actors: End User
- Preconditions:Application must be downloaded
- Steps: User opens and uses application
- Post-conditions: End User will be able to discover the device's precise relative location
- Exceptions: Application isn't downloaded or used properly

Chapter 4

Activity Diagram

The following is an activity diagrams that show the flow of decisions being made by the framework as it encounters a location request from an app build on top of it.

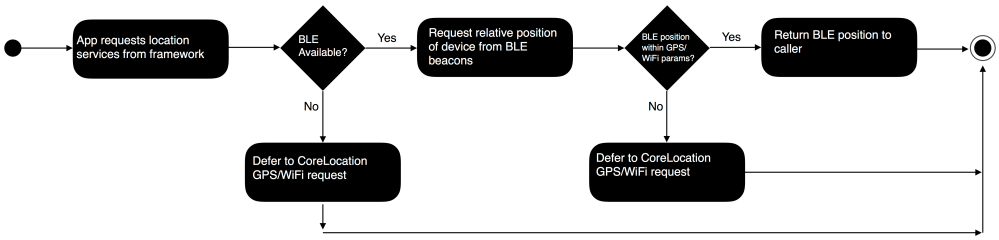

Figure 4.1: Activity Diagram

Chapter 5

Conceptual Model

The conceptual model displays examples of our framework in use by an app. This specific on is taking place in a mall.

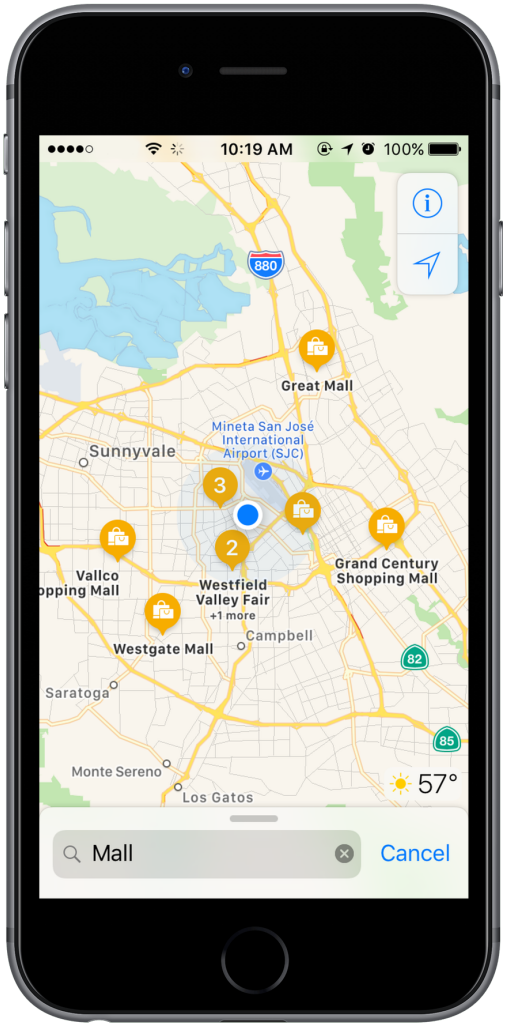

Figure 5.1: Discovering Beacons View
This is an example of what it would look like while the device is finding bluetooth beacons

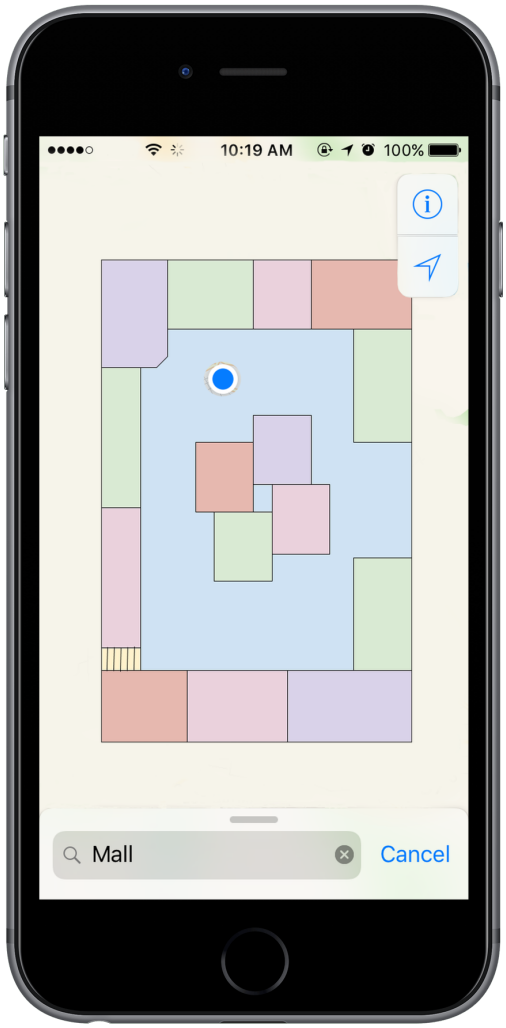

Figure 5.2: Location View
This is an example of what it would look like while the device is displaying relative location

Chapter 6

Architectural Diagram

The following architectural diagram shows that the framework conceptually is build on top of the iOS CoreLocation service.The admin web app will only have to be reachable by the framework during the initial phase of the system installation or when major changes are made to the system.

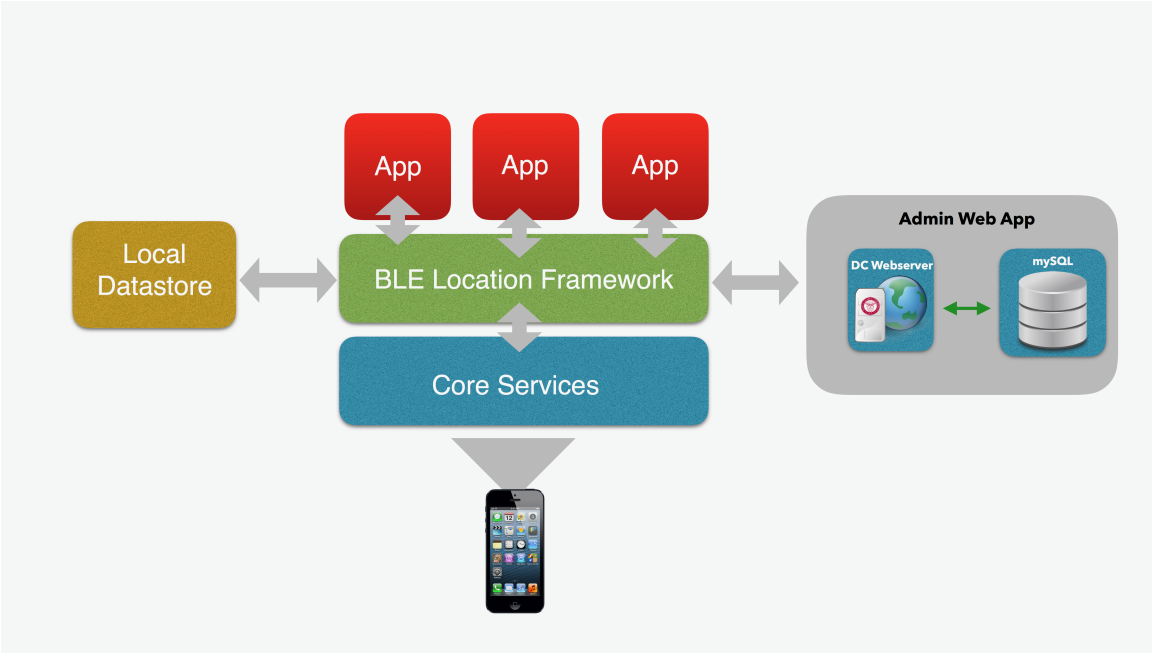

Figure 6.1: Architectural Diagram

Chapter 7

Technologies Used

Framework:

- Swift - Swift is the most modern language used for producing native iOS apps.It is more readable and flexible than Obj-C, making it a better choice for this project. It is also backwards compatible with Obj-C and C++ libraries so it does not limit the usage of open source material in the system.
- C++ - C++ is a very portable and widely used language that can interface with many other languages easily.For this reason C++ will be used for sections of the framework that are purely mathematical algorithms.This will allow for the core algorithms relating to Trilateration and Fingerprinting Heuristics to be easily ported to other platforms for further testing and development.

Admin Webservice:

- HTML/CSS - Will be used for formatting and styling of admin webpage.
- Javascript and JQuery - Javascript will be used to add responsiveness and to provide underlying front end logic. JQuery will be used to simply DOM manipulation of admin webpage, and to simplify AJAX calls.
- PHP - Will be used on the back end of the admin web service to sanitize inputs and interface with the database.
- mySQL - Will be used for the datastore of the admin web service.

Network:

- iBeacon - the iBeacon protocol will be used in all broadcasting beacons

Chapter 8

Design Rationale

PLs: Swift and a modern and well supported language that will provide both the native support necessary and the robustness and optimization to support heavy parallel computation.

C++ is industry standard for mathematical algorithms that require a large amount of customization and testing.C++ is also well supported by other languages that are able to build a wrapper around core code in order to provide front end flexibility.

HTML, CSS, Javascript, PHP, and mySQL are all industry standard and well tested and documented.This makes them the obvious choice to implement the web interface for the admin portal.

Architecture: Building the framework on top of the existing CoreLocation framework in iOS provides a level of abstraction to developers who want to use precise location services without needing to know the details of that hardware system is providing the data.While conceptually the framework is built on top of CoreLocation, in practice it would be more accurate to say it extends CL.This means that developers who do want to fine tune the way they use the BLE Positioning Framework will be able to do so.

Hosting the configuration portal for the system as a web service provided over the Internet have potential drawbacks (needing and Internet connection at some points limits certain geographical regions).However, the initial positioning data of given beacon sets needs to be stored somewhere, and a central web server is much safer than a distributed system which leans more heavily on the competence of the end user.

Chapter 9

Test Plan

- Unit Testing: Most testing will be done by our developers using manual white box testing. Automated testing will not be used because it would not give us the desired results and will take as much time setting up as it would do to just test manually.

  We will start off with General Feature Testing wherein we will code a particular feature and test it for it's functionality. As the Unit Testing model recommends, every time we write a new feature, we will test it for its functionality ensuring minimization of errors. This method will help us break our system into smaller, more manageable parts, the testing of which will be relatively simpler. It will also help us make these small parts ready to be used as a part of the bigger system.

- Regression Testing: As we will progress with the development of the system, we will begin implementing Regression Tests on top of the general feature testing. Unlike what we will do in Unit Testing, in Regression Testing we will write and test a particular feature and test the system. Once everything works fine, we will add the next feature and then retest the system from the beginning, instead of just testing the new feature.

  Simply put, regression testing is testing the system in its entirety every time a change or addition is made to it to make sure the already implemented features are not be affected by the new features in an undesirable way.

  Regression testing will help us efficiently manage our system whenever a change will be made.

- Ad hoc Testing: Ad hoc testing is performed without a plan of action. It will be performed by improvisation where the testers will find bugs by any means that seemed appropriate. Ad hoc tests will help us spot out any strange bugs that could arise in the operation of the system.

Chapter 10

Risk Analysis

The Risk Analysis table defines a potential set of risks that our group could face, as setbacks to timely progression towards our finished project. For each risk, there are two potential consequences, a probability value 0 to 1, a severity value 0 to 10, an impact value (impact=probability*severity), and potential mitigation strategies.

Risk	Consequences	Probability	Severity	Impact	Mitigation Strategies
Planned implementation failed to work.	Mid cycle decisions/changes are required	0.4	7	2.8	Research and careful planning
Loss of code	Significant setback in timeline and extra work	0.1	10	1	Use Git and Githut for everything. Don't forget to push code.
Scheduling conflicts	Pace of project completion slowed/delayed. Communication is not fluid.	0.25	4	1	Clean communication. Schedule meetings far in advance.
Illness	Work distribution shifted unevenly. Project pace slowed	0.3	6	1.8	Get plenty of sleep. Don't cram for schoolwork
Over budget	Forced to cut features and and cut automation of test processes, slowing down the project	0.4	6	2.4	Research all system components and their individual costs

Figure 10.1: Risk Analysis Table

The two most potentially impactful risks are if the planned implementation of the system failed to provided good results, and if the project runs over budget.By coding in a modular style that provides low coupling, the system can be modified if necessary without having to start from scratch.Additionally, by buying most line items in the budget up front, we can be sure that the costs do not balloon as the project progresses.

Chapter 11

Technologies Used

Framework:

- Swift - Swift is the most modern language used for producing native iOS apps.It is more readable and flexible than Obj-C, making it a better choice for this project. It is also backwards compatible with Obj-C and C++ libraries so it does not limit the usage of open source material in the system.
- C++ - C++ is a very portable and widely used language that can interface with many other languages easily.For this reason C++ will be used for sections of the framework that are purely mathematical algorithms.This will allow for the core algorithms relating to Trilateration and Fingerprinting Heuristics to be easily ported to other platforms for further testing and development.

Admin Webservice:

- HTML/CSS - Will be used for formatting and styling of admin webpage.
- Javascript and JQuery - Javascript will be used to add responsiveness and to provide underlying front end logic. JQuery will be used to simply DOM manipulation of admin webpage, and to simplify AJAX calls.
- PHP - Will be used on the back end of the admin web service to sanitize inputs and interface with the database.
- mySQL - Will be used for the datastore of the admin web service.

Network:

- iBeacon - the iBeacon protocol will be used in all broadcasting beacons

Chapter 12

Ethical Analysis

- Ethical
  - There are no real ethical questions because there is no outcome that could have an immoral aspect.
- Social
  - This could have an impact on society because it could decrease the amount of time people spend at malls or other places due to more efficient navigation of a building. People might spend less time in public, will less likely run into people at the mall or other places, and overall reduce person-to-person contact. It could also reduce the amount of impromptu purchases from businesses because people will not have to walk by unnecessary stores.
- Political
  - There are no real political questions because this will not affect government at any level besides maybe easier navigation in their buildings.
- Economic
  - The cost of development will be relatively low because iBeacons are inexpensive, easy to maintain, and the framework can be reproduced as many times as you want for no extra cost.
- Health and Safety
  - There are no real Health and Safety questions because bluetooth is safe for public use.
- Manufacturability

  - iBeacons are an existing product and the framework requires no manufacturing.

- Sustainability
  - iBeacons' batteries can be sustained for several years on average so they won't need to be replaced very often.

- Environmental Impact
  - There will be minimal waste but once an iBeacon dies it will need to be recycled.

- Usability
  - Our framework will be easy to use for developers to help them design excellent applications.

- Lifelong learning
  - This project will help our group to continually learn how to gain knowledge outside a traditional lecture.

- Compassion
  - This project has a compassionate side because it will help reduce the stress for developers creating location based apps and reduce stress for users when navigating inside buildings.

Chapter 13

Development Timeline

The Development Timeline is a graphical representation of all the tasks that are needed for the completion of the project. This graphical model is known as Gantt Chart.

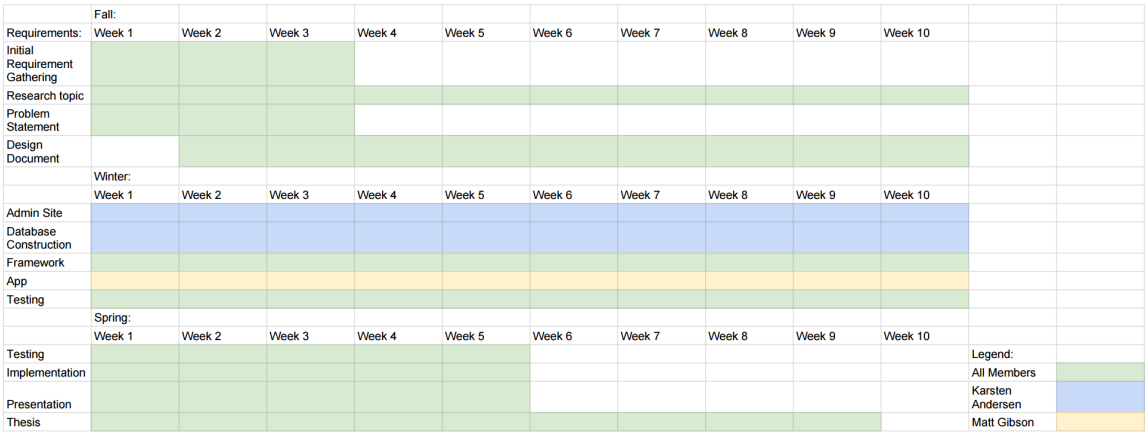

Figure 13.1: Gantt Chart

The table columns in the Gantt Chart lists the set time periods over which the project is distributed in. For our purpose, we have divided the project across ten weeks for each quarter. The rows of the table in the chart lists the various tasks that need to completed. The intersection of the rows and columns in the Gantt Chart would provide data about the team member who will be working on a respective task and when a particular task is due for submission or presentation. The Gantt Chart proves to be an efficient visual method of tracking the different tasks of the project and helps staying on schedule.