|  |  |  |
| --- | --- | --- |
| Design Document EasiCAB | |  |
| This is a design document for EASiCAB Project. This document elaborate about the design and flow architecture of the whole project. | EASiCAB for EASi | |

Contents

[2. Revision history: 3](#_Toc458527018)

[3. Introduction 3](#_Toc458527019)

[3.1 Purpose of this document 3](#_Toc458527020)

[3.2 Scope 3](#_Toc458527021)

[3.3 Reference 3](#_Toc458527022)

[3.4 Definitions and acronyms 4](#_Toc458527023)

[4. Use Case 4](#_Toc458527024)

[Introduction 4](#_Toc458527025)

[4.1 Use Case Diagram objects 4](#_Toc458527026)

[4.2 Actors in Use Case Diagram 6](#_Toc458527027)

[4.2.1 Admin 6](#_Toc458527028)

[4.2.2 Easi Employees 6](#_Toc458527029)

[5. Flow Chart 10](#_Toc458527030)

[6. Architecture of Application 12](#_Toc458527031)

[7. Data Design 12](#_Toc458527032)

[8. SNAPSHOTS 13](#_Toc458527033)

# 

# Revision history:

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 25 July   2015 | 0.1 | Initial Draft, Mobile application | Nitesh Kumar |

.

# Introduction

The Software Design Document is a document to provide documentation which will be used to aid in software development by providing the details for how the software should be built. Within the Software Design Document are narrative and graphical documentation of the software design for the project including use case models, sequence diagrams, collaboration models, object behavior models, and other supporting information.

## Purpose of this document

The purpose of the Software Design Document is to provide a description of the design of a system fully enough to allow for software development to proceed with an understanding of what is to be built and how it is expected to build. The Software Design Document provides information necessary to provide description of the details for the software and system to be built.

## Scope

This Software Design Document is for a base level system which will work as a proof of concept for the use of building a system. This provides a base level of functionality to show feasibility for large scale production use. This Software Design is focused on the base level system and critical parts of the system. For this particular Software Design Document, the focus is placed on generation of the documents and modification of the documents. The system will be used in conjunction with other pre-existing systems and will consist largely of a document interaction facade that abstracts document interactions and handling of the document objects

## Reference

For Document provided by Customer, check the link in SVN:

[https://asi-storage07.blr.allegisindia.com/svn/iPlanter/trunk/Customer\_Spec](https://asi-storage07.blr.allegisindia.com/svn/iPlanter/trunk/Customer_Spec )

## Definitions and acronyms

|  |  |
| --- | --- |
| **Keyword** | **Definitions** |
| EASiCAB | Our project name. |

# Use Case

## Introduction

Use case diagram is a behavioral UML diagram type and frequently used to analyze various systems. They enable you to visualize the different types of roles in a system and how those roles interact with the system. They are also used to gather a usage requirement of a system. Depending on your requirement you can use that data in different ways. Below are few ways to use them.

* **To identify functions and how roles interact with them** – The primary purpose of use case diagrams.
* **For a high level view of the system** – Especially useful when presenting to managers or stakeholders. You can highlight the roles that interact with the system and the functionality provided by the system without going deep into inner workings of the system.
* **To identify internal and external factors** – This might sound simple but in large complex projects a system can be identified as an external role in another use case.

## 

## 4.1 Use Case Diagram objects

Use case diagrams consist of 4 objects.

* Actor
* Use case
* System
* Package

##### **4.1.1 Actor**

##### Actor in ause case diagram is **any entity that performs a role** in one given system. This could be a person, organization or an external system and usually drawn like skeleton shown below.

Description: Description: Actor

##### **4.1.2 Use Case**

##### A use caserepresents a function or an action within the system**.** It’s drawn as an oval and named with the function.



##### **4.1.3 System**

##### System is used to define the scope of the use case and drawn as a rectangle. This an optional element but useful when your visualizing large systems. For example you can create all the use cases and then use the system object to define the scope covered by your project. Or you can even use it to show the different areas covered in different releases.



##### **4.1.4 Package**

##### Package is another optional element that is extremely useful in complex diagrams. Similar to [class diagrams](http://creately.com/diagram-type/class-diagram), packages are used to group together use cases. They are drawn like the image shown below.

[](http://static3.creately.com/blog/wp-content/uploads/2014/03/Package1.png)

## 4.2 Actors in Use Case Diagram

### 4.2.1 Admin

Admin is the end users who will use the app for assigning cab drivers to employees and can also monitor the locations of employee and cab on Google map. He/she can also update the cab drivers, time shift and route of the cab accordingly.

### 4.2.2 Easi Employees

They are the real user of this mobile app. Easi employees can register, signup and after login he/she will be able to see the driver details and the location of the cab on Google map. Employees can see the notifications about cab and after taking the cab , they can check in and while leaving cab can check out. He/she can also call or message driver and admin as well.

**4.2.3 Cab Drivers**

They will be able to sign up and register via the mobile app. After login he will be able to see his details and the location of his next destination in Google map. Google map will show the route to the drivers as well. They can accept the check in notifications of employees. He wil be able to call and receive calls from employees and admin.



Fig : 3.3



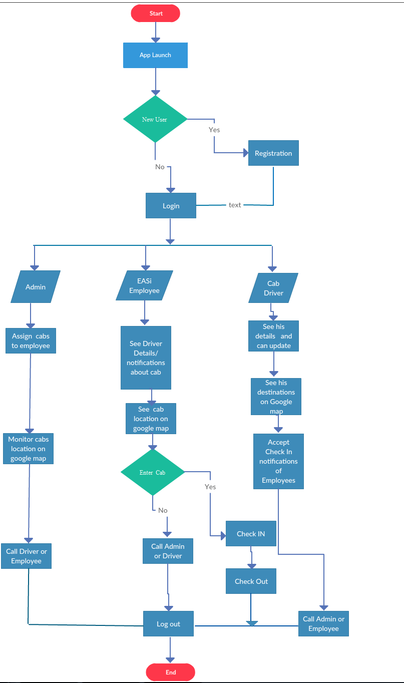
Fig 3.4



Fig : 3.5

# Flow Chart

A **flowchart** is a type of diagram that represents an algorithm, workflow or process, showing the steps as boxes of various kinds, and their order by connecting them with arrows. This diagrammatic representation illustrates a solution model to a given problem.

  
 Fig: 4.1

**Overview:**

As per diagram **Fig.4.1:** represent the control flow of EASiCAB mobile app **.**

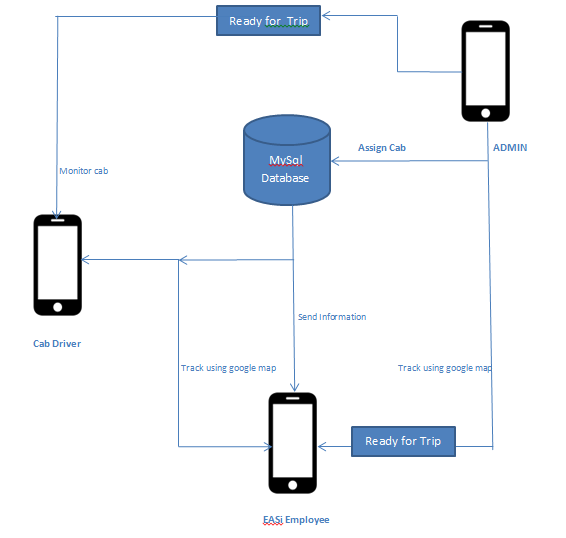
On first time launch of the app, it the user is new, then he/she will register his details into the app. If he/she has already registered, then will login to the app. According to user description during registration, he will automatically go to the admin or employee or driver screen.

In admin screen, user can assign cabs to the employees and then can monitor the cab location on Google map. He can call driver or employee if required.

In employee screen, user can see driver details and notifications about cab. They can see cab location on Google map. If he/she can enter the cab, then they can be able to check in and while leaving the cab, they can check out. They can also call driver or admin if required.

In cab driver screen, he can see his details and his destination on Google map. Google map will show him the route as well. He can accept the check in notifications of employees and can call admin or employee if required.

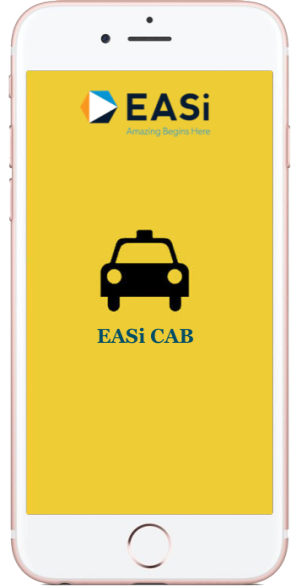
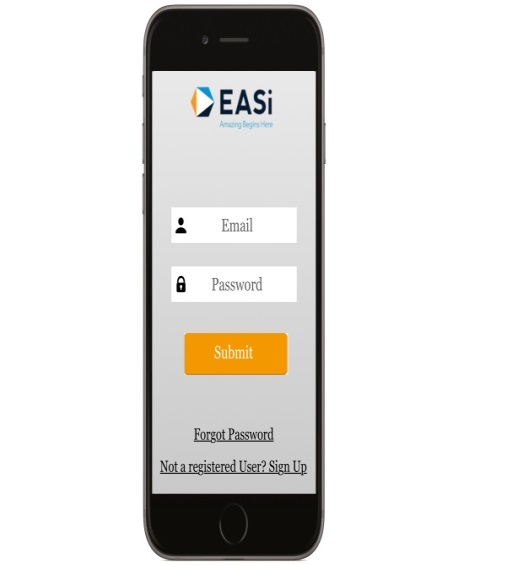
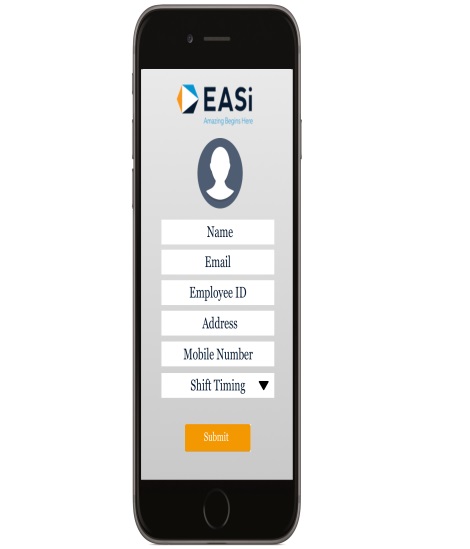
# Architecture of Application



# Data Design

Refer link [EASi Cabs Database Design v1.0.doc](EASi%20Cabs%20Database%20Design%20v1.0.doc)

# SNAPSHOTS

­­­­