**As Partial Fulfillment of:**

**MSc (IT) - Master of Science In Information Technology**

**Submitted To:**

**Atmiya Institute of Technology and Science – 2017**

GUIDED BY: Prakash Gujarati

PREPARED BY: Dipen Parmar

**ACKNOWLEDGEMENT**

I would like to highlight the fact that this project is the product of many contributors. First, I’d like to thank Professor **Prakash Gujarati** who accepted to work with me on my Project idea, supervising my work.

Second, I would like to thank the stack overflow community and Google for providing tremendous help to the elaboration of this project.

**ABSTRAT**

The Lara-POS Restaurant Management software is a capstone project that aims towards developing an all-in-one application that addresses the various problems and challenges faced by high-end restaurant owners today.

In order to achieve this goal, this project addresses various aspects of the modern Restaurant Management System in India.

All in all, this projects main aim is to reduce the time overhead in high-end management restaurants by providing an alternative to the traditional management system based on physical record keeping and paper work.

**PROJECT PROFILE**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | **PROJECT:** | Lara-POS Restaurant Management. | | **OBJECTIVE:** | Pendding | | **INSTITUTE:** | Atmiya Institute of Technology and Science | | **FRONT END:** | Bootstrap 4, jQuery, | | **BACK END:** | PHP 7.1.3 , Laravel 5.6 , MySql | | **DOCUMENTATION TOOL:** | Ms Word, Ms Excel | | **DEVELOPED BY:** | Dipen Parmar | | **GUIDED BY:** | Prakash Gujarati | | **SUBMITTED TO:** | Msc-it (AITS) | | **-** |  | | **-** |  | |  |
|  |  |

**INDEX**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr** | **Particulars** | | **PageNo** |
|  |  |  |  |
| **1** | **Title Page** | | - |
| **2** | **ACKNOWLEDGEMENT** | | - |
| **3** | **ABSTRAT** | | - |
| **3** | **PROJECT PROFILE** | | - |
| **4** | **UML DIAGRAMS** | | () |
| 4.1 | Use Case | - |
|  | 4.2 | Activity Diagram | - |
|  | 4.3 | Wire Frame | - |
|  | 4.4 | User Interface | - |
|  | 4.5 | Data Dictionary | - |
|  | 4.6 | Data Flow Diagram (DFD) | - |
|  | 4.7 | E-R Diagram |  |
| **5** | **Report Softcopy** | | () |
| **6** | **----------------------------** | | - |
|  | 6.1 | User Interface | - |
|  | 6.2 | Database Structure, Migrations |  |
| **7** | **Implementation** | | () |
|  | 7.1 | Errors and Solutions |  |
|  | 7.2 | Challenging Tasks / Problems |  |

**HARDWARE REQUIREMENTS**

**Technical Details Hardware (**Requirements**)**

|  |  |
| --- | --- |
| **Particulars** | **Minimum Hardware**  **Requirements** |
| Processor Brand | Intel, Amd etc. |
| Processor Type | Core 2 Duo |
| Processor Speed | 1.0 GHz |
| RAM Size | 1 GB |
| Memory Type | DDR2 |
| Hard Drive Size | 30 GB |
| Hard Drive Interface | Any |

**Technical Details Software (**Requirements**)**

|  |  |
| --- | --- |
| **Particulars** | **Minimum Software**  **Requirements.** |
| Operating System | Windows, Linux, Mac, any |
| Browser | • Chrome\* 36+ • Edge\* 20+ • Mozilla Firefox 31+ |

**USE CASE**

A use case describes how a user uses a system to accomplish a particular goal. A use case diagram consists of the system, the related use cases and actors and relates these to each other to visualize: what is being described? (system), who is using the system? (actors) and what do the actors want to achieve? (use cases), thus, use cases help ensure that the correct system is developed by capturing the requirements from the user's point of view.

----

**What is a use case diagram?**

In the Unified Modeling Language (UML), a use case diagram can summarize the details of your system's users (also known as actors) and their interactions with the system. To build one, you'll use a set of specialized symbols and connectors. An effective use case diagram can help your team discuss and represent:

* Scenarios in which your system or application interacts with people, organizations, or external systems
* Goals that your system or application helps those entities (known as actors) achieve
* The scope of your system

**When to apply use case diagrams?**

A use case diagram doesn't go into a lot of detail—for example, don't expect it to model the order in which steps are performed. Instead, a proper use case diagram depicts a high-level overview of the relationship between use cases, actors, and systems. Experts recommend that use case diagrams be used to supplement a more descriptive textual use case.

UML is the modeling toolkit that you can use to build your diagrams. Use cases are represented with a labeled oval shape. Stick figures represent actors in the process, and the actor's participation in the system is modeled with a line between the actor and use case. To depict the system boundary, draw a box around the use case itself.

UML use case diagrams are ideal for:

* Representing the goals of system-user interactions
* Defining and organizing functional requirements in a system
* Specifying the context and requirements of a system
* Modeling the basic flow of events in a use case

Use case diagram components

To answer the question, "What is a use case diagram?" you need to first understand its building blocks. Common components include:

**Actors:** The users that interact with a system. An actor can be a person, an organization, or an outside system that interacts with your application or system. They must be external objects that produce or consume data.

**System:** A specific sequence of actions and interactions between actors and the system. A system may also be referred to as a scenario.

**Goals:** The end result of most use cases. A successful diagram should describe the activities and variants used to reach the goal.

**Use case diagram symbols and notation**

The notation for a use case diagram is pretty straightforward and doesn't involve as many types of symbols as other UML diagrams.

* **Use cases:** Horizontally shaped ovals that represent the different uses that a user might have.
* **Actors:** Stick figures that represent the people actually employing the use cases.
* **Associations:** A line between actors and use cases. In complex diagrams, it is important to know which actors are associated with which use cases.
* **System boundary boxes:** A box that sets a system scope to use cases. All use cases outside the box would be considered outside the scope of that system. For example, Psycho Killer is outside the scope of occupations in the chainsaw example found below.

**System**:  
 Draw your system's boundaries using a rectangle that contains use cases. Place actors outside the system's boundaries.



**Use Case**:  
 Draw use cases using ovals. Label the ovals with verbs that represent the system's functions.

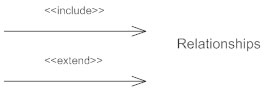
****

**Actors:** Actors are the users of a system. When one system is the actor of another system, label the actor system with the actor stereotype.



**Relationships:**

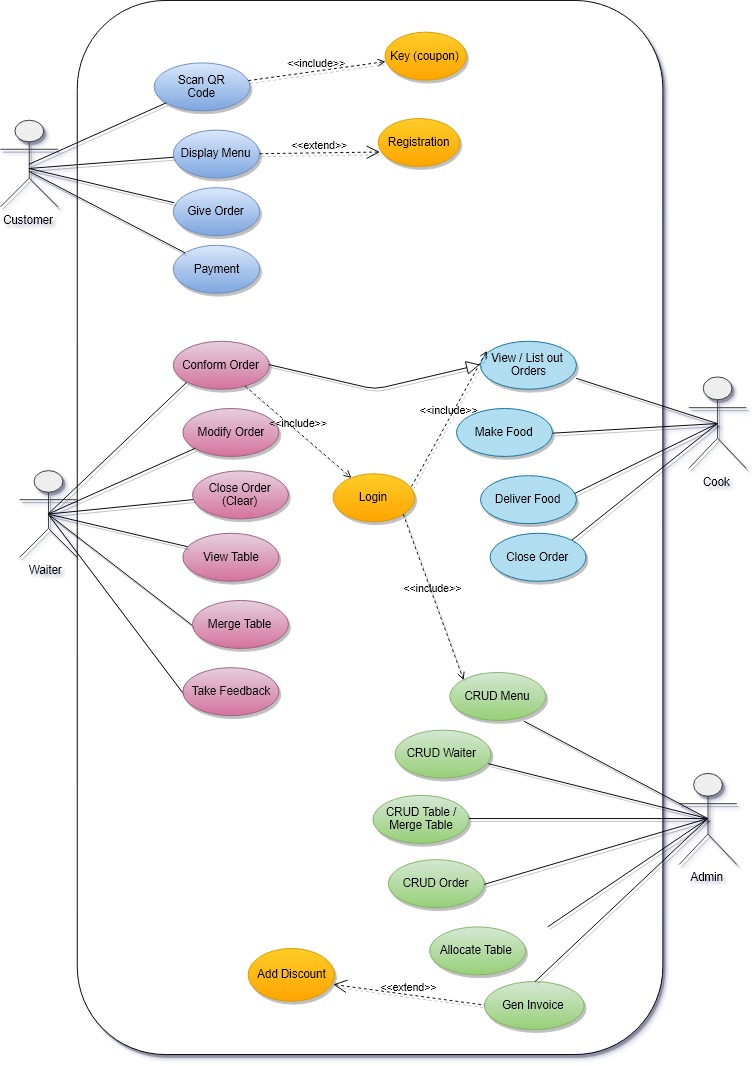
Illustrate relationships between an actor and a use case with a simple line. For relationships among use cases, use arrows labeled either "uses" or "extends." A "uses" relationship indicates that one use case is needed by another in order to perform a task. An "extends" relationship indicates alternative options under a certain use case.



Src 1: <https://www.lucidchart.com/pages/uml-use-case-diagram>

Src 2: <https://www.smartdraw.com/use-case-diagram/>

* **Use case diagram for Restaurant management system:**



Drawing Tool Used- : **Draw.io**

**Draw**.**io** is free online diagram software for making flowcharts, process diagrams, org charts, UML, ER and network diagrams.

Website: <https://www.draw.io/>

**ACTIVITY DIAGRAM**

**Activity diagram**: is another important behavioral diagram in UML diagram to describe dynamic aspects of the system. Activity diagram is essentially an advanced version of flow chart that modeling the flow from one activity to another activity.

## When to Use Activity Diagram?

Activity Diagrams describe how activities are coordinated to provide a service which can be at different levels of abstraction. Typically, an event needs to be achieved by some operations, particularly where the operation is intended to achieve a number of different things that require coordination, or how the events in a single use case relate to one another, in particular, use cases where activities may overlap and require coordination. It is also suitable for modeling how a collection of use cases coordinate to represent business workflows

1. Identify candidate use cases, through the examination of business workflows
2. Identify pre- and post-conditions (the context) for use cases
3. Model workflows between/within use cases
4. Model complex workflows in operations on objects
5. Model in detail complex activities in a high level activity Diagram

* Activity Diagram Notation Summary

| Symbol | Name | Description |
| --- | --- | --- |
| start Symbol | Start symbol | Represents the beginning of a process or workflow in an activity diagram. It can be used by itself or with a note symbol that explains the starting point. |
| activity Symbol | Activity symbol | Indicates the activities that make up a modeled process. These symbols, which include short descriptions within the shape, are the main building blocks of an activity diagram. |
| connector Symbol | Connector symbol | Shows the directional flow, or control flow, of the activity. An incoming arrow starts a step of an activity; once the step is completed, the flow continues with the outgoing arrow. |
| joint Symbol | Joint symbol/ Synchronization bar | Combines two concurrent activities and re-introduces them to a flow where only one activity occurs at a time. Represented with a thick vertical or horizontal line. |
| fork Symbol | Fork symbol | Splits a single activity flow into two concurrent activities. Symbolized with multiple arrowed lines from a join. |
| decision Symbol | Decision symbol | Represents a decision and always has at least two paths branching out with condition text to allow users to view options. This symbol represents the branching or merging of various flows with the symbol acting as a frame or container. |
| note Symbol | Note symbol | Allows the diagram creators or collaborators to communicate additional messages that don't fit within the diagram itself. Leave notes for added clarity and specification. |
| send signal Symbol | Send signal symbol | Indicates that a signal is being sent to a receiving activity. |
| receive signal Symbol | Receive signal symbol | Demonstrates the acceptance of an event. After the event is received, the flow that comes from this action is completed. |
| shallow history pseudostate symbol | Shallow history pseudostate symbol | Represents a transition that invokes the last active state. |
| option loop symbol | Option loop symbol | Allows the creator to model a repetitive sequence within the option loop symbol. |
| flow final symbol | Flow final symbol | Represents the end of a specific process flow. This symbol shouldn’t represent the end of all flows in an activity; in that instance, you would use the end symbol. The flow final symbol should be placed at the end of a process in a single activity flow. |
| condition text | Condition text | Placed next to a decision marker to let you know under what condition an activity flow should split off in that direction. |
| end symbol | End symbol | Marks the end state of an activity and represents the completion of all flows of a process. |

­