1. **Introduction**

**1.1 Purpose**

In System development systems design plays the vital role. The purpose of System Designis to create a technical solution that satisfies the functional requirements for the system. The Functional Specification produced during System Requirements Analysis is transformed into a physical architecture.

System components are distributed across the physical architecture, usable interfaces are designed and prototyped and technical specifications are created for the application developers, enabling them to build and test the system.

Design document helps in better understanding of the functionality of the system. These diagrams show how the data is processed at different stages and the sequences of events that take place during some user interaction with the system. This document gives a brief idea of the functionality and the way the different users of the project interact with each other.

**1.2 Overview**

This document contains use case diagram, sequence diagram and deployment diagrams of lunch decider. These diagrams show the behavior of the system and the common attributes and operations provided by different objects, actors.

**1.3 Tool (Star UML) Used to Design**

Star UML is a free UML modeling tool that is simple and has flexible integrated set of options to design any kind of diagram. Start Star UML, and select ’Empty Project’ at the startup. Then in the Model Explorer add a new model by right-clicking on the ’untitled’ thing. After that you can create all kinds of UML diagrams by right-clicking on the model.

**2. Design Diagrams**

There are several types of design diagrams and has been divided into various categories such as,

i) Static Diagrams: It emphasizes the static structure of the system using objects, attributes, operations and relationships.

1. Use case diagrams
2. Class diagrams
3. Component diagrams
4. Deployment diagrams

ii) Dynamic Diagrams: It emphasizes the dynamic behavior of the system by showing collaborations among objects and changes to the internal states of objects.

1. Object diagrams
2. Sequence diagrams
3. Collaboration diagrams
4. State chart diagrams
5. Activity diagrams

**2.1 Use case Diagram**

Use case is used to describe the interaction between external users of the system i.e., actors and the system itself. Use case diagram is a set of use cases. The “include” relationship shown in figure 1 in UML is a stereotype which a describes “relationship between two use cases, implying that the behavior of the included use case is inserted into the behavior of the including use case"[2].

Actor may have common behavior and also the way it interacts with the actions in the use case, those actors can be combined and described them in a general way and it is represented using “generalization” relationship. The square box in figure 1 is the system boundary of our application.

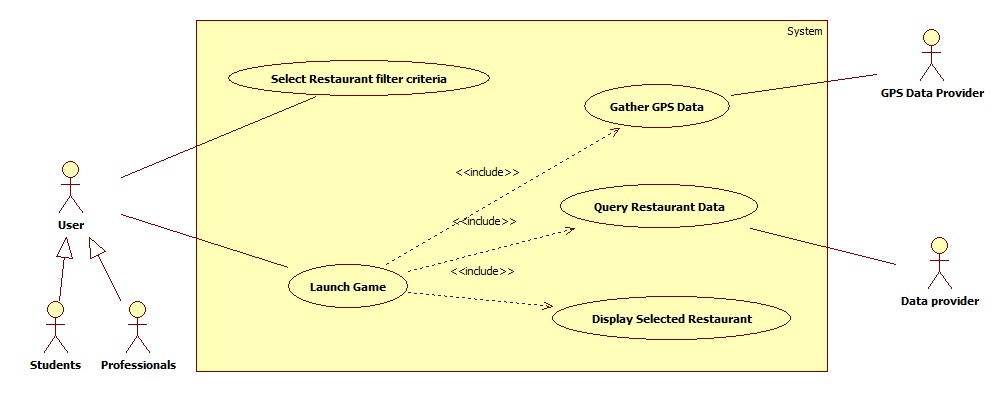
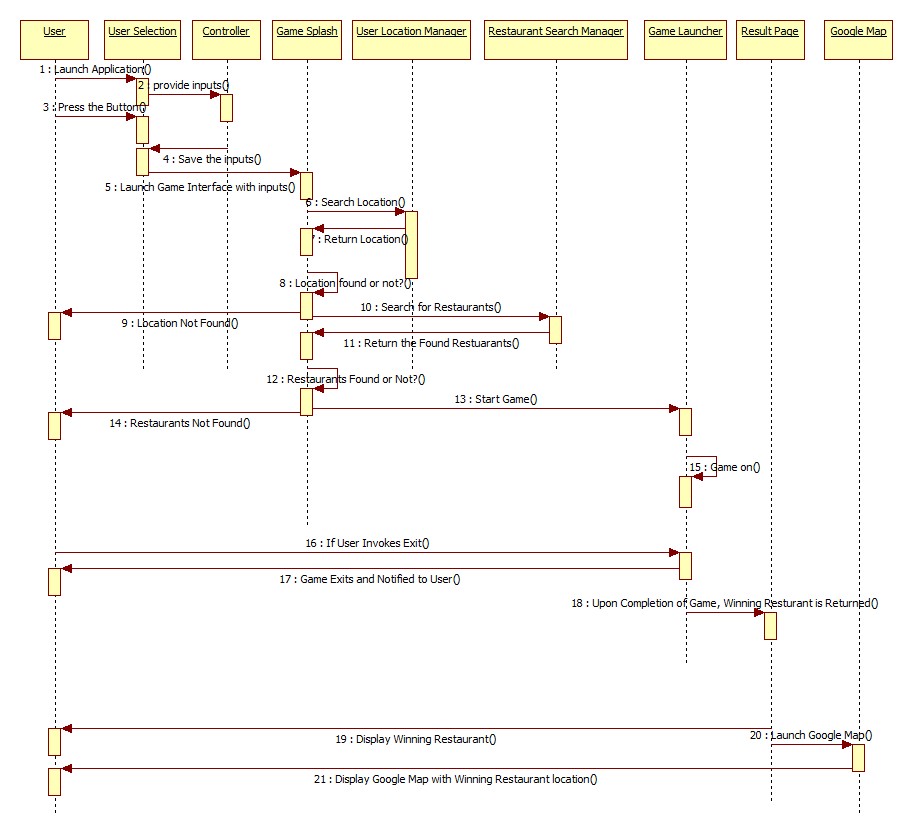


Figure 1: Use Case Diagram for Lunch Decider Application

GPS (Global Positioning System) data provider is used to find the user location in interacts with gather GPS data. Data provider is not any external database it is our own database with is kept in server. An association exists whenever an actor is involved with an interaction described by a use case and is denoted by solid line.

**2.2 Sequence Diagram**

Sequence diagram is part of the interaction diagrams show the way that the objects in the system interact with one another and the order of interaction by messages flow. In sequence diagram the vertical lines represent the “life line” i.e., the objects life time and the horizontal arrows represents the messages exchanged between objects in the order in which they occur.

****Figure 2: Sequence Diagram for Lunch Decider Application

User, user selection, controller, game splash, user location manager, restaurant search manager, game launcher, result page, google map are the objects in the sequence diagram and messages numbering from 1 to 21 are the interaction between various objects in the order they occurred.

**3. References**

[1]Stephen R. Schach, *Object-Oriented and Classical Software Engineering*. 8th. McGraw-Hill, 2010.

[2] http://en.wikipedia.org/wiki/Use\_case\_diagram#Associations

[3] http://en.wikipedia.org/wiki/Sequence\_diagram