Southeast University (SEU)
Department of Computer Science & Engineering (CSE)

online food shop System

Information System Design
&
Software Engineering Lab(**CSE 346.3**)

Submitted By:
Group Name : The supersonics

Name: PARVES KHAN ID:2020000000036

Name: Md. Mahadi Hasan ID:2020000000089

Name: sharmin sultana ID:2020000000123

Introduction

- The online food delivery system is a web based system where people can order food from restaurant . This system enables to deliver food to the customer by using web site. This system will show all the available food, locations,food availability and customers can choose food of their choice.

Goals of the project

1. Build a online food shop system.
2. The system will store details of all the added food and order the food by the choice of the customer.
3. food can be added and removed by the customer.
4. Customer, shop owner, delivery man will be registered at the system.
5. Customer can order food that suit their needs.
6. To provide a food delivery service anytime and anywhere.

Case study

online food shop system is a small business that order food from online. It provide a food service within shop open time and nearby shop area.It is a new system .It was inspired by food panda system. At first it is a small company with a handful of employees working on small project. The system will store details of all the added food and food details along with the desired destination and fee. Customer, shop owner, Delivery man are registered at the system.Customer order food that suit their needs. It provide a food service specific time and location. Customer visit the system & select the number of food, the type of food & necessary information.

But they're experiencing problems with maintaining the system properly as well as providing a user friendly interface. Sometimes they fail to provide service late at night and remote areas due to lack of service. They not having a 24/7 service. Instability in the website there is

no option for bargaining. So that we are discussing Provide a user friendly web based interface for the users.Provide an enriched database to the system.

Problem Statement

Existing food rental systems are experiencing problems with maintaining the system properly as well as providing a user friendly interface. Lack of chance to communicate between the food shop owner and customer. Sometimes they fail to provide service late at night and remote areas due to lack of service.There is always an objection among the customer about the food quality and sometimes over price.

Issues

1. The system interface is difficult to use.
2. The system requires a lot more information regarding the food shop owner, deliveryman and thus the customer can go through it and communicate with them.

Objectives

1. Provide a user friendly web based interface for the users.
2. Provide an enriched database to the system. So that, we can have the information we need.
3. Make the website more organize.
4. Provide a easy food ordering system.

Feasibility

Technical Feasibility: Availability of technology and resource along with enough manpower will help us to complete the system.

Economical Feasibility: Because of using most of the resources for free we can develop the system at a very low cost. Thus,it will enable us to provide the software at a very reasonable price to the users.

Operational Feasibility: Proper number Human Resources are available to make the system operative and the system interface is very user friendly to operate.

Cost-Benefit Analysis:

Tangible Cost:

Cost of programmer's time and cost of resources are the tangible costs.

Intangible Cost:

Losing the reputation of being first.

Declining company image.

Ineffective decision making

Tangible Benefit:

Quick access to the information.

The system will be more fluent.

Intangible Benefit:

Increase of company image.

Become a strong competitor in the market.

Stakeholders or Users:

=> Admin

=> customer

=> shop owner

=> delivery boy

=> Developers

Functional Grouping According to the Types of Users:

Admin: Admin will monitor the whole system like who can access the system, who is logging into the website etc. They will report any problem like bug, security issue to the developers.

customer: customer can create new user account, login to the website.They can see food information, price, delivery time.

Deliveryman: Deliveryman can see customer information, the route. They can also create and login to their account.

shop owner: shop owner will work as bridge between customer and deliveryman. They will manage food for the customer.

Developers: Developers will continuously work in the system for development and maintenance. They will fix the bug that will be reported to them from the admin panel, users. They will add new features to the system and keep updating

Non-functional Requirements(NFRs):

Performance:

1. The system should execute any command within 3 seconds.

2. The system should be able to access Google map.

3. The system should be able to give service to 70 users at a time.

Security:

The system must be well protected so that no unauthorized person can access the system.

The system should not allow unauthorized person to collect user's information.

The system must make sure user's information will remain safe and secure.

Availability:

The system should be available 24/7

Maintainability:

The system should be under maintenance one time in every month

Implementation:

The system will be implemented using HTML,CSS,PHP,MySQL

Standard:

The default language of the system should be English.

The system should be authorized by the corresponding department of the government.

Manageability:

The system should be built up in such a way that it can be easily used

<u>Overall Constraints</u>:

=> The system has been made only for web platform; android and iOS version is not available.

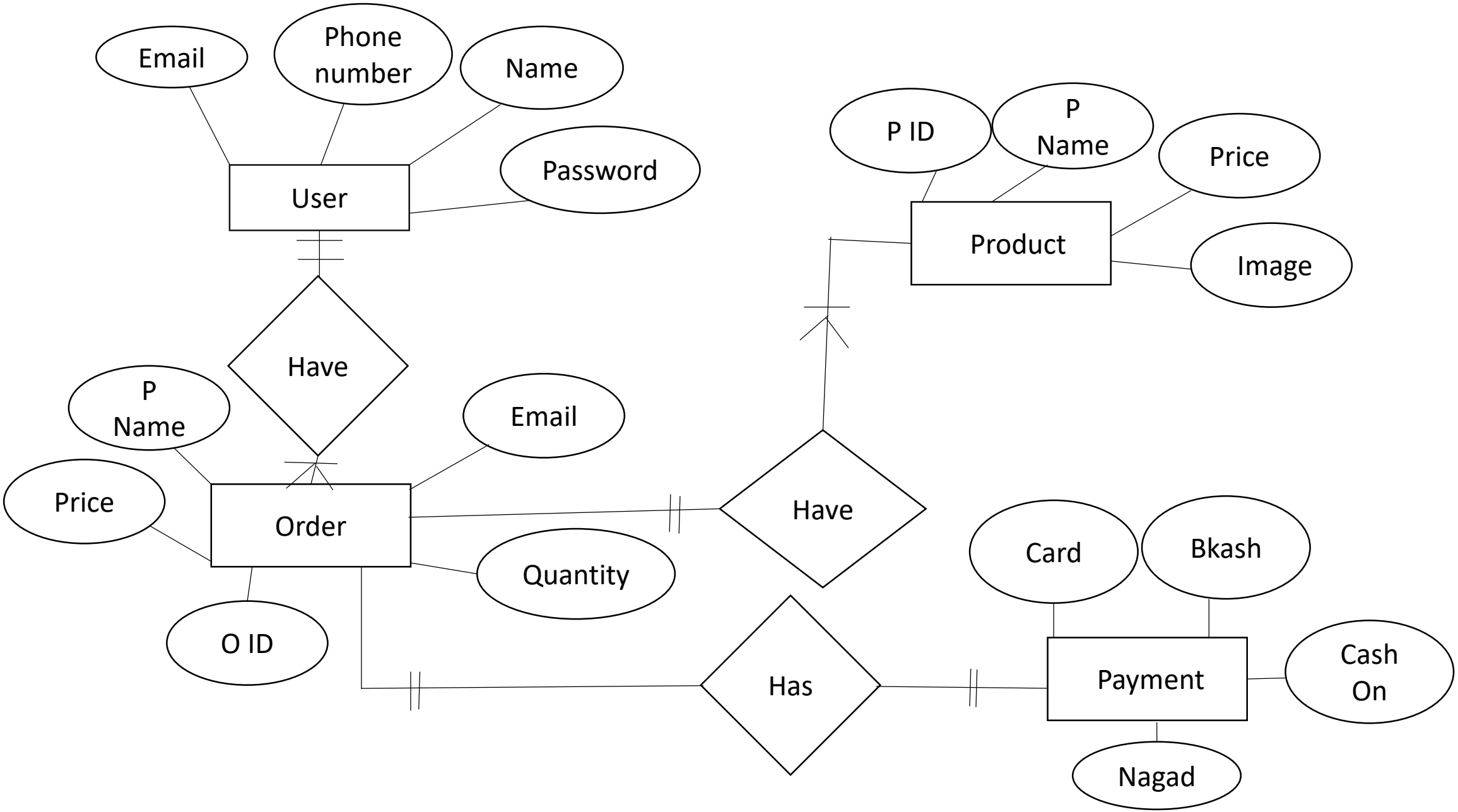
=> The system is only made for food shop.

Relationship Matrix

	User	Order	Product	Payment
User		√		
Order	√		√	√
Product		√		
Payment		√		

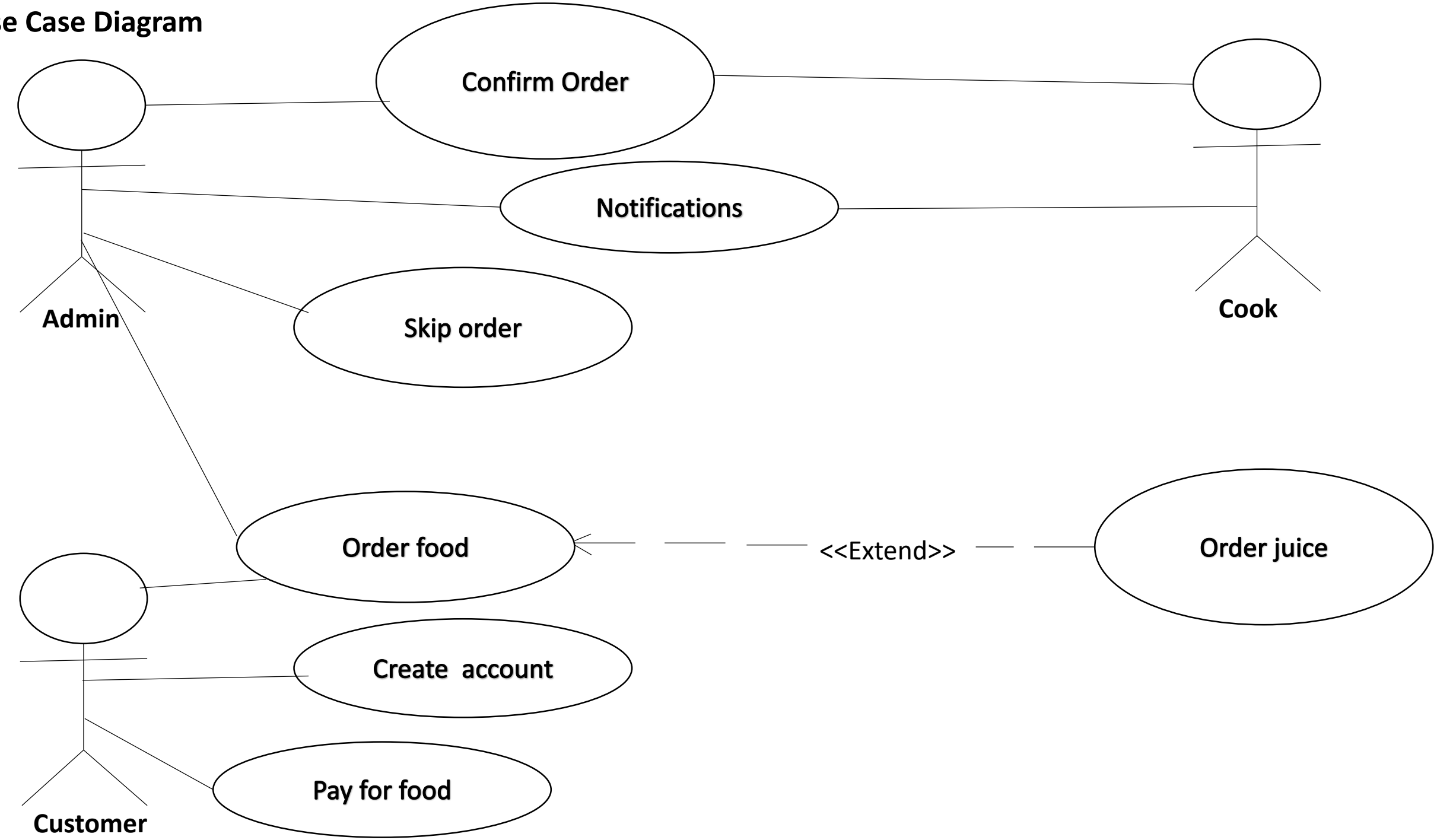
Entity-Relationship Diagram (ERD): Entity Relationship Diagram, also known as ERD, ER Diagram or ER model, is a type of structural diagram for use in database design. An ERD contains different symbols and connectors that visualize two important information: The major entities within the system scope, and the inter-relationships among these entities. Components of an ERD: ▪ Entity: An entity may be any object, class, person or place. In the ER diagram, an entity can be represented as rectangles. Consider an organization as an example manager, product, employee, department etc. can be taken as an entity. ▪ Attribute: The attribute is used to describe the property of an entity. Eclipse is used to represent an attribute. For example, id, age, contact number, name, etc. can be attributes of a student. ▪ Relationship: A data relationship is a natural association that exists between one or more entities. ▪ Cardinality: Defines the number of occurrences of one entity for a single occurrence of the related entity. The process has ten steps: • Identify Entities • Find Relationships • Draw Rough ERD • Fill in Cardinality • Define Primary Keys • Draw Key-Based ERD • Identify Attributes • Map Attributes • Draw fully attributed ERD • Check Results

Entity Relationship Diagram



Use Case Diagram: A use case diagram is a visual summarization of interactions and relationships within a system. These diagrams show a very broad view of a system. They may show systems in computer software, businesses or customer experiences. A use case diagram shows a model scenario in which individuals interact with a system using a series of specialized symbols and connectors. Components of Use Case Diagram (UCD) are as follows: ● Actors: The users of the system. Refer to someone or something that has a goal in using the system. Represent particular role played by one user or a group of external users/customers in the system! Can be not just person but also organizations or another systems/sub-systems, must be linked to at least one use case, also may interact with more than one use cases. A use case may involve one or more actors. Some examples of actors can be Customer, Student, Passenger, Bank, Server etc. ● Use Cases: Functionalities of the system. Refers to a unit of complete and useful functionality that business or system provides to its users. Describes how external user interacts with a system to achieve a desired result, that is, how the system is going to be used. Captures a specific functional requirement (behavior) of the system. Can be thought of a sequence of transactions in the system. Provides developers with a view of what the users want. ● System Boundary: Overall Scope of the system. It is potentially the entire system as defined in the requirements document. Represented using a rectangle with all the use Cases inside the rectangle and all the actors outside the rectangle. ● Communication Lines: Relationships between various other components.

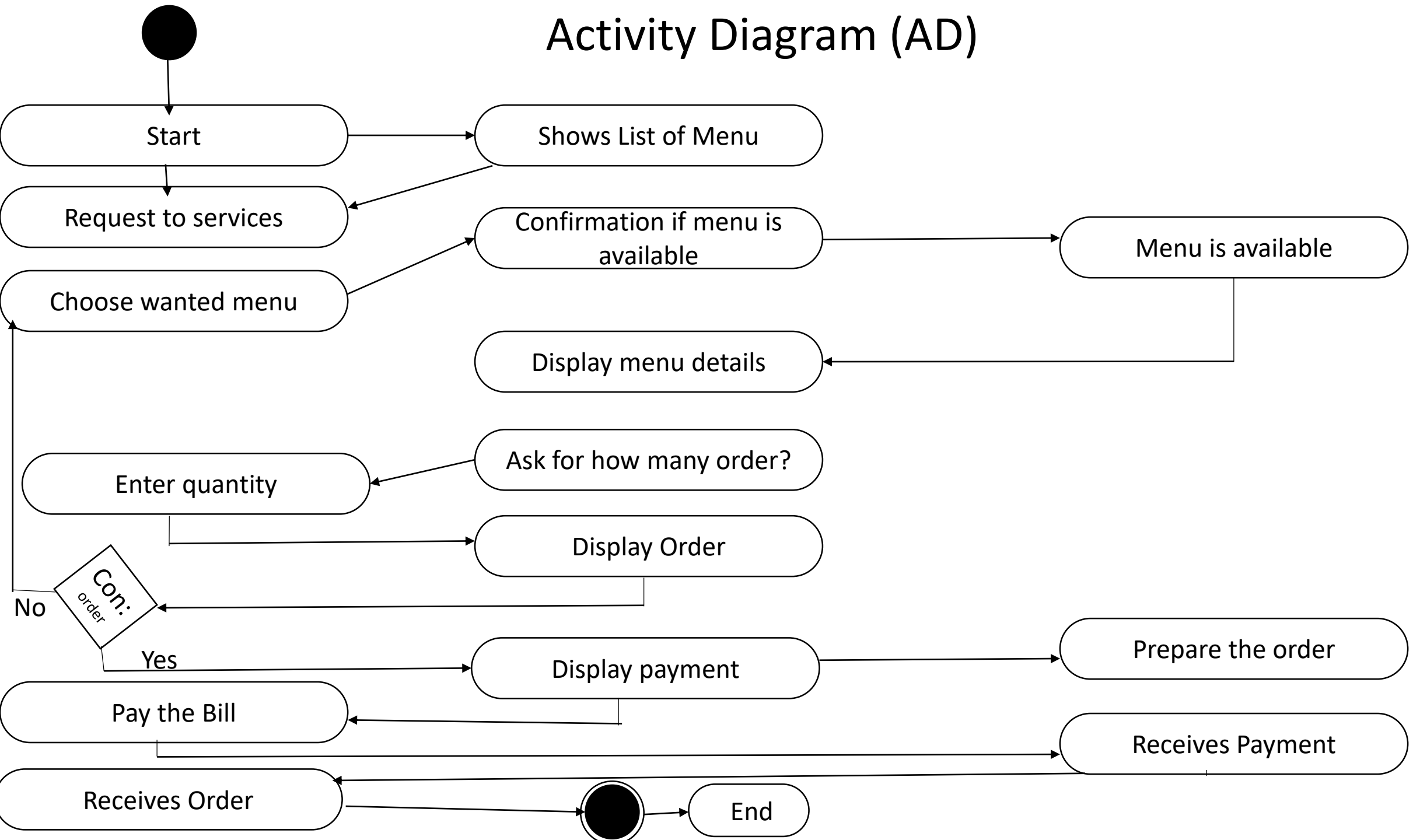
Use Case Diagram



Activity Diagram: Activity diagram is essentially an advanced version of flowchart that models the flow from one activity to another activity

Activity Diagram Notation Summary: Notation Description UML Notation Activity Is used to represent a set of actions Action A task to be performed Control Flow Shows the sequence of execution Object Flow Show the flow of an object from one activity (or action) to another activity (or action). Initial Node Portrays the beginning of a set of actions or activities Activity Final Node Stop all control flows and object flows in an activity (or action) Object Node Represent an object that is connected to a set of Object Flows Decision Node Represent a test condition to ensure that the control flow or object flow only goes down

Activity Diagram (AD)

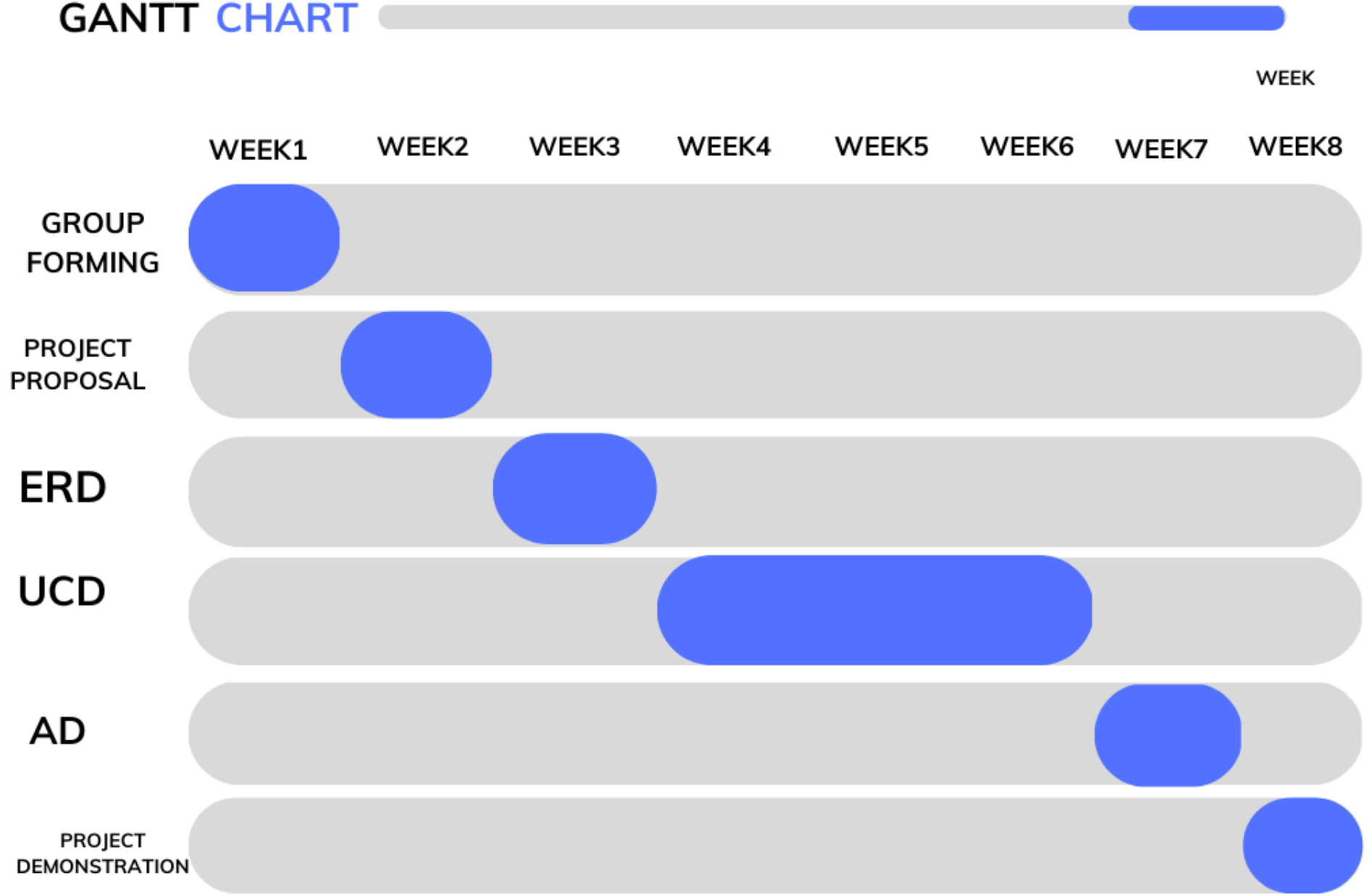


GANTT Chart: GANTT Chart is a type of bar chart which represents the project schedule of a project. It was named after its inventor Henry Gantt. We can determine the project duration from the chart. It is one of the most useful tools of showing activities displayed against the time. Characteristics of GANTT Chart: ● Each task is represented by a bar. ● The position of each bar reflects the start and end time of the corresponding activity ● The length of each bar represents the duration of the corresponding activity ● The x-axis of the chart represents the duration of an activity ● The y-axis represents the activities of a project To construct a GANTT chart, we need the project schedule which will consist of Activity, Predecessor and Duration columns. Here in our project's perspective, there are six activities. These are: Group forming, Project proposal submission, Entity Relationship Diagram (ERD), Use Case Diagram (UCD) , Activity Diagram(AD) and Project demonstration. Each activity except Group forming has their own predecessor. Like project group forming has to be done before submitting the project proposal. There is also a column titled 'Duration' in the schedule indicated how many units of time (for this project we have calculated the time in week unit) required to complete the activity. For example, for designing and submitting the Use Case Diagram (UCD) we needed 3 weeks. To construct the GANTT chart, we plotted the activities in the y-axis and duration in the x-axis. Then we wrote drew bars of every activity according to their duration. We followed the precedence of every activity to draw the GANTT chart.

Activity	Precedence	Duration
Group Forming	-------	1
Project proposal	Group Forming	1
ERD	Project proposal	1
UCD	ERD	3
AD	UCD	1
Project Demonstration	AD	1

Project proposal includes Introduction ,Goal of the project Case Study, Problem statement, Issues/Limitations, Object Feasibility analysis, Cost benefit analysis, Types of stakeholders, Functionality Grouping to the type of users, NFRs, Constraints.

GANTT CHART


WEEK
WEEK1
WEEK2
WEEK3
WEEK4
WEEK5
WEEK6
WEEK7
WEEK8
GROUP FORMING
PROJECT PROPOSAL
ERD
UCD
AD
PROJECT DEMONSTRATION