CIS 330
Project Preliminary Design and Detailed Design

System of Interest
The System of Interest (SOI) is a system that is the focus of a project. It is also defined as: "the system-of-interest contains system elements, system element interconnections, and the environment in which they are placed."

System Design:
As shown in Figure 6-3 of the textbook, there are five main activities that comprise System Design. The five activities are:

1. Describe the environment.
2. Design the application components.
3. Design the user interface.
4. Design the database.
5. Design the software classes and methods.

Describe the Environment
Most new information systems must fit into an already existing technology environment. Hence this activity is more focused towards defining and understand the environment rather than designing the technology infrastructure from scratch. Essentially there are two key elements that must be defined:

- External systems, meaning other systems that the new system must interface with, and
- Technology architecture, meaning the complete environment of hardware and software that supports the new system.

Design the Application Components
Designing the components refers to defining the set of components that will be needed and how they all interconnect together. Many new systems consist of pre-packaged components and in-house programmed components. These components can range from small special purpose modules, such as one to calculate sales tax or to generate a GUID, to complete subsystems, such as a payment processing subsystem. Decisions must be made to define the components, decide whether to buy or build each component, what programming language is required, etc.

Design the User Interface
This is one of the most critical and yet one of the most difficult design activities. It is critical because, to the users, the interface is the system. That is all they see and interact with. A poorly designed user interface can lead to a system that is used inefficiently or is avoided by users altogether. UI/UX is also difficult because of the many types of devices and options that must be supported. Not only are there many devices and options, but new ones appear on the horizon frequently. User interface design is also complex and it cannot be done in isolation as it requires heavy user involvement throughout.

Many techniques can be used to do user interface design, including models, mockups, samples, story boards, and so forth.

Design the Database
Information systems usually contain a major database component. If the database is not designed correctly, then the information requirements will not be met correctly. Database design is a fairly mature activity with standard rules and approaches. If the data analysis was done thoroughly, then the actual design of the database can proceed rapidly.

Design the Software Classes and Methods
This activity is often thought of as detailed design. Depending on the complexity of the system either a minimal set of models can be generated, or a thorough and complex design can be done. The primary objective of this activity is to define and design the classes and methods that are needed to carry out each use case. A complex design will utilize such models as design class diagrams and interaction diagrams.

Deliverables:

For this assignment, the team will submit one document consisting of six sections:

1. Define the system of interest:

Define the system of interest with a description and a diagram. You must use Microsoft Visio or a similar UML modeling program to create your System of Interest. Example:

"For this research, the SOI consists of three separate subsystems. The first subsystem is the Wait Time Forecasting Subsystem composed of the forecasting model(s) that will be used to predict the customer waiting times for full-service restaurants. The second subsystem is a database for the entire system. It will be hosted on the cloud and it will consist of a database that will store all of the data used by the system. The third subsystem is the mobile application that will be used by several types of system users to access and interact with the system. The mobile application may be written as a native app for both the iOS and Android operating systems, which will require that the subsystem be coded twice. The app will allow users to request services from the system back end such as wait time forecasts, restaurant reviews, and directions among other things. Customers will also provide data to the system including their reviews of restaurants, payment data, and requests for additional information. The system of interest is shown below:"

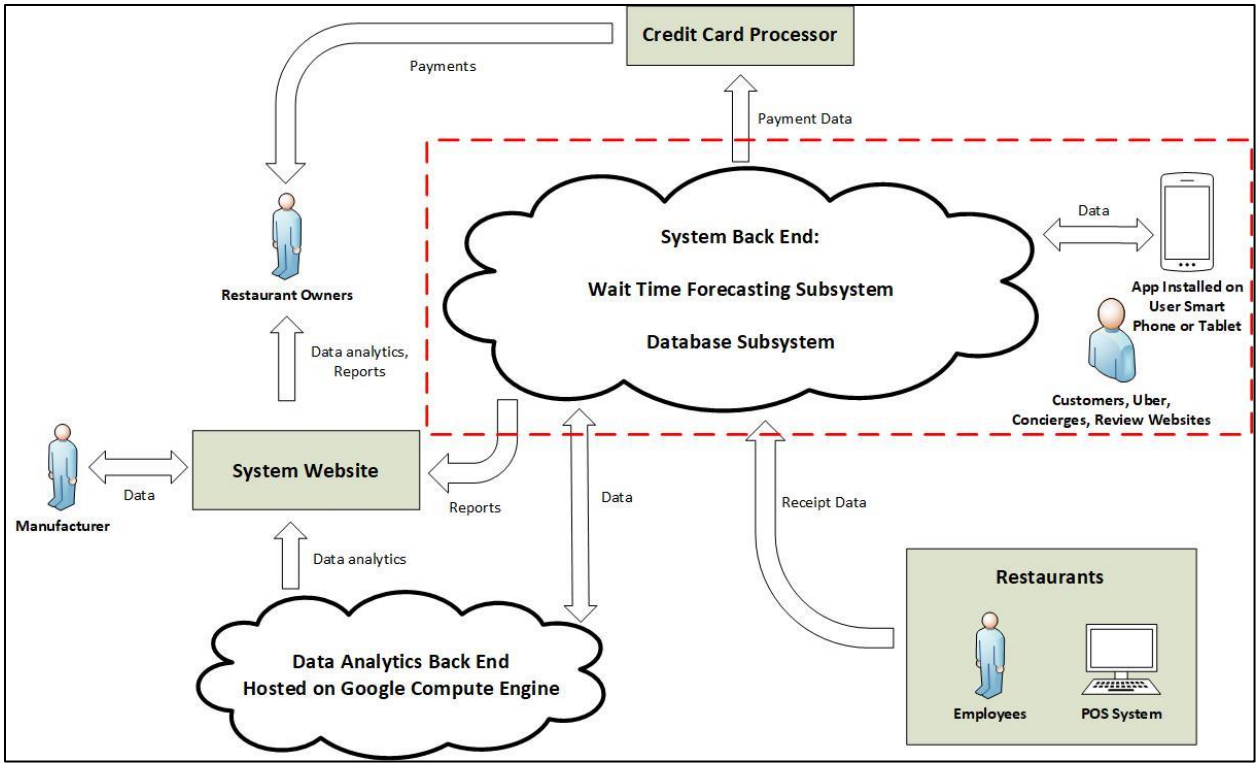


Figure 1. System of interest is inside of the red dashed box

2. Describe the environment:

Provide a narrative describing the environment of the proposed system and answer how will this system interact with other systems and with the organization's existing technologies. Show how the proposed system will interact with other systems and with the organization's existing technologies? See page 164 of the textbook for additional help with this deliverable.

3. Design application components

Identify the key parts of the proposed system and how they will interact when the system is deployed. Is the proposed system a stand-alone system or does it need to be divided into subsystems? Research to see if any components can be purchased.

Various models are used to design the application components, including package diagrams, deployment diagrams, and component diagrams. You must use Microsoft Visio or a similar UML modeling program to create all diagrams. Examples are given in Figure 6-5 of the textbook. Chapters 7 and 13 of the text cover this process in more detail.

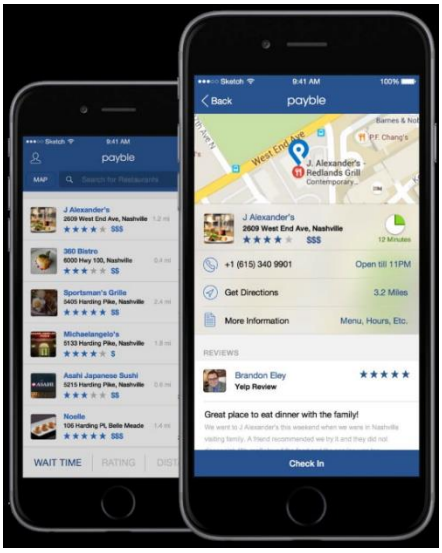
4. Design the user interface:

The user interface will be one of the more difficult portions of this assignment. This process shows how the users will interact with the proposed system. Your answer can include text, images, and even videos. To be able to have a meaningful display on desktops, laptops, and mobile devices some care will need to be given in designing the user interface. Tasks will include design sessions with some users for each type of device. Perhaps even some trial prototypes will need to be built to test the effectiveness of different screen layouts. Chapter 8 covers this process in greater detail. It is strongly recommended that you use an integrated

development environment (IDE) such as Eclipse or NetBeans to create your user interface(s). (Since the CIS 315 course uses Eclipse as the IDE for the class, it may be helpful for you to use it to create your UI screens for this class as well.) Both Eclipse and NetBeans are available on the computers on the second floor of the Computing and Mathematics building. Also, you may use any IDE you wish to do this portion of the work.

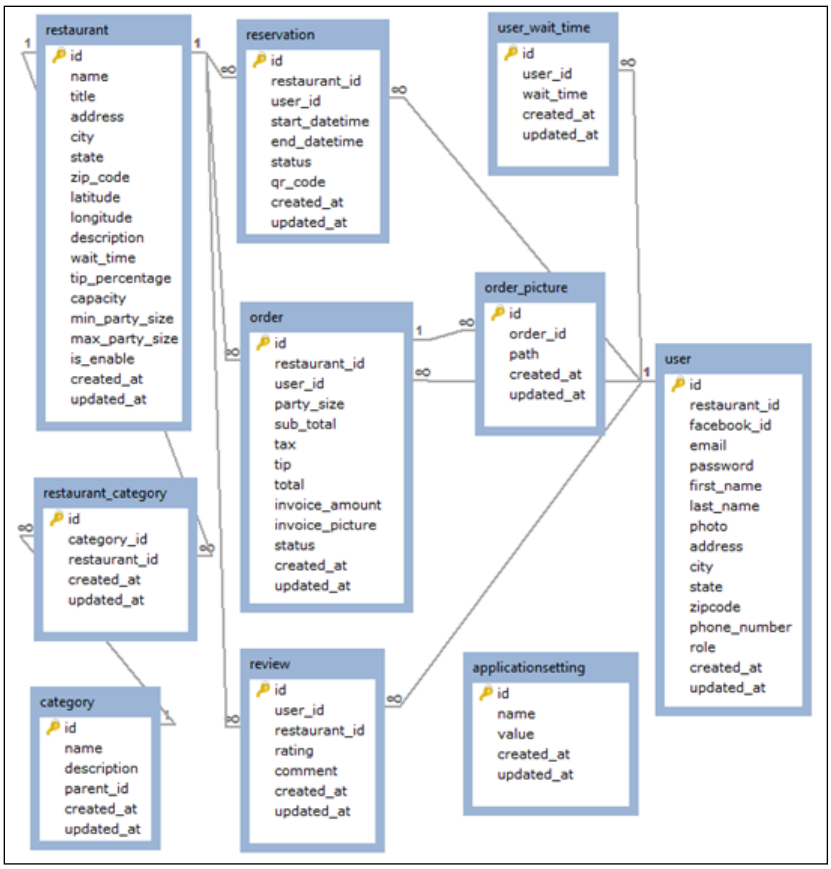
Note that the user interface is not required to have working functionality. Providing a diagram showing how all of the different UI screens are connected together is helpful here. Two examples are shown below:

An example of a completed user interface for a mobile device is shown below:



5. Design the database:
This process shows how data is captured, structured, and stored for later use by the system. Design of the database will require defining the tables based on the classes and relationships in the data model. Also, the various indexes and searching options must be identified. Other tasks include defining the attribute characteristics and foreign keys. Chapter 9 covers database design. This deliverable can easily be done in MySQL (preferred) or Microsoft Access. Both of these can be used by UNA students at no cost.

An example of a database structure is shown below:



6. Design the software classes and methods:
This process seeks to determine what internal structure for each application component will ensure efficient construction, rapid deployment, and reliable operation. Design will follow the normal design steps of specifying the code structure and the methods. Tasks will include creating class diagram(s), system sequence diagrams, and state machine diagrams. Look at figures 6-8, 5-7, and 4-33 respectively. Again, you must use Microsoft Visio (or similar) to create all diagrams.

Submit your work in two ways:

Your work from Word, Excel, and Visio for all of the steps above should be combined into a single document with everything clearly labeled and neatly organized.

1. Save your document in your GitHub repository. If your GitHub repository is public, submit a link to it in the assignment drop box in the course's Canvas site. If your GitHub is private, add the professor to your repository by inviting user rddavisjr.

2. Submit a single pdf, .doc, or docx file to the assignment drop box in Canvas. (This will be used for ABET purposes.)