**Software Engineering Project Report**



**Requirement Document for Reserve Your Table**

**Prepared by**

**Juan Vazquez**

**Nguyen Hoa Pham**

**Gurleen Kaur**

**Jujhar Singh**

**for use in CS 440**

**at the**

**University of Illinois Chicago**

**February 2021**

***REMOVE OR REPLACE ALL TEXT IN RED ITALICS BEFORE SUBMITTING REPORT***

**Table of Contents**

[List of Figures 7](#_Toc529138678)

[List of Tables 8](#_Toc529138679)

[I Project Description 9](#_Toc529138680)

[1 Project Overview 9](#_Toc529138681)

[2 The Purpose of the Project 9](#_Toc529138682)

[2a The User Business or Background of the Project Effort 9](#_Toc529138683)

[2b Goals of the Project 9](#_Toc529138684)

[2c Measurement 9](#_Toc529138685)

[3 The Scope of the Work 9](#_Toc529138686)

[3a The Current Situation 10](#_Toc529138687)

[3b The Context of the Work 10](#_Toc529138688)

[3c Work Partitioning 11](#_Toc529138689)

[3d Competing Products 12](#_Toc529138690)

[4 The Scope of the Product 12](#_Toc529138691)

[4a Scenario Diagram(s) 12](#_Toc529138692)

[4b Product Scenario List 12](#_Toc529138693)

[4c Individual Product Scenarios 12](#_Toc529138694)

[5 Stakeholders 12](#_Toc529138695)

[5a The Client 12](#_Toc529138696)

[5b The Customer 13](#_Toc529138697)

[5c Hands-On Users of the Product 13](#_Toc529138698)

[5d Maintenance Users and Service Technicians 13](#_Toc529138699)

[5e Other Stakeholders 13](#_Toc529138700)

[5f User Participation 13](#_Toc529138701)

[5g Priorities Assigned to Users 13](#_Toc529138702)

[6 Mandated Constraints 14](#_Toc529138703)

[6a Solution Constraints 14](#_Toc529138704)

[6b Implementation Environment of the Current System 14](#_Toc529138705)

[6c Partner or Collaborative Applications 14](#_Toc529138706)

[6d Off-the-Shelf Software 14](#_Toc529138707)

[6e Anticipated Workplace Environment 14](#_Toc529138708)

[6f Schedule Constraints 14](#_Toc529138709)

[6g Budget Constraints 15](#_Toc529138710)

[7 Naming Conventions and Definitions 15](#_Toc529138711)

[7a Definitions of Key Terms 15](#_Toc529138712)

[7b UML and Other Notation Used in This Document 15](#_Toc529138713)

[7c Data Dictionary for Any Included Models 15](#_Toc529138714)

[8 Relevant Facts and Assumptions 15](#_Toc529138715)

[8a Facts 15](#_Toc529138716)

[8b Assumptions 15](#_Toc529138717)

[II Requirements 16](#_Toc529138718)

[9 Product Use Cases 16](#_Toc529138719)

[9a Use Case Diagrams 16](#_Toc529138720)

[9b Product Use Case List 17](#_Toc529138721)

[9c Individual Product Use Cases 17](#_Toc529138722)

[10 Functional Requirements 19](#_Toc529138723)

[11 Data Requirements 19](#_Toc529138724)

[12 Performance Requirements 20](#_Toc529138725)

[12a Speed and Latency Requirements 20](#_Toc529138726)

[12b Precision or Accuracy Requirements 20](#_Toc529138727)

[12c Capacity Requirements 20](#_Toc529138728)

[13 Dependability Requirements 21](#_Toc529138729)

[13a Reliability Requirements 21](#_Toc529138730)

[13b Availability Requirements 21](#_Toc529138731)

[13c Robustness or Fault-Tolerance Requirements 21](#_Toc529138732)

[13d Safety-Critical Requirements 22](#_Toc529138733)

[14 Maintainability and Supportability Requirements 22](#_Toc529138734)

[14a Maintenance Requirements 22](#_Toc529138735)

[14b Supportability Requirements 22](#_Toc529138736)

[14c Adaptability Requirements 23](#_Toc529138737)

[14d Scalability or Extensibility Requirements 23](#_Toc529138738)

[14e Longevity Requirements 23](#_Toc529138739)

[15 Security Requirements 23](#_Toc529138740)

[15a Access Requirements 24](#_Toc529138741)

[15b Integrity Requirements 24](#_Toc529138742)

[15c Privacy Requirements 24](#_Toc529138743)

[15d Audit Requirements 24](#_Toc529138744)

[15e Immunity Requirements 25](#_Toc529138745)

[16 Usability and Humanity Requirements 25](#_Toc529138746)

[16a Ease of Use Requirements 25](#_Toc529138747)

[16b Personalization and Internationalization Requirements 25](#_Toc529138748)

[16c Learning Requirements 26](#_Toc529138749)

[16d Understandability and Politeness Requirements 26](#_Toc529138750)

[16e Accessibility Requirements 26](#_Toc529138751)

[16f User Documentation Requirements 27](#_Toc529138752)

[16g Training Requirements 27](#_Toc529138753)

[17 Look and Feel Requirements 27](#_Toc529138754)

[17a Appearance Requirements 27](#_Toc529138755)

[17b Style Requirements 28](#_Toc529138756)

[18 Operational and Environmental Requirements 28](#_Toc529138757)

[18a Expected Physical Environment 28](#_Toc529138758)

[18b Requirements for Interfacing with Adjacent Systems 28](#_Toc529138759)

[18c Productization Requirements 29](#_Toc529138760)

[18d Release Requirements 29](#_Toc529138761)

[19 Cultural and Political Requirements 29](#_Toc529138762)

[19a Cultural Requirements 29](#_Toc529138763)

[19b Political Requirements 30](#_Toc529138764)

[20 Legal Requirements 30](#_Toc529138765)

[20a Compliance Requirements 30](#_Toc529138766)

[20b Standards Requirements 30](#_Toc529138767)

[21 Requirements Acceptance Tests 31](#_Toc529138768)

[21a Requirements – Test Correspondence Summary 31](#_Toc529138769)

[21b Acceptance Test Descriptions 31](#_Toc529138770)

[III Design 32](#_Toc529138771)

[22 Design Goals 32](#_Toc529138772)

[23 Current System Design 32](#_Toc529138773)

[24 Proposed System Design 32](#_Toc529138774)

[24a Initial System Analysis and Class Identification 32](#_Toc529138775)

[24b Dynamic Modelling of Use-Cases 32](#_Toc529138776)

[24c Proposed System Architecture 32](#_Toc529138777)

[24d Initial Subsystem Decomposition 33](#_Toc529138778)

[25 Additional Design Considerations 33](#_Toc529138779)

[25a Hardware / Software Mapping 33](#_Toc529138780)

[25b Persistent Data Management 33](#_Toc529138781)

[25c Access Control and Security 33](#_Toc529138782)

[25d Global Software Control 33](#_Toc529138783)

[25e Boundary Conditions 34](#_Toc529138784)

[25f User Interface 34](#_Toc529138785)

[25g Application of Design Patterns 34](#_Toc529138786)

[26 Final System Design 34](#_Toc529138787)

[27 Object Design 34](#_Toc529138788)

[27a Packages 35](#_Toc529138789)

[27b Subsystem I 35](#_Toc529138790)

[27c Subsystem II 35](#_Toc529138791)

[27d etc. 35](#_Toc529138792)

[IV Project Issues 35](#_Toc529138793)

[28 Open Issues 35](#_Toc529138794)

[29 Off-the-Shelf Solutions 35](#_Toc529138795)

[29a Ready-Made Products 35](#_Toc529138796)

[29b Reusable Components 35](#_Toc529138797)

[29c Products That Can Be Copied 36](#_Toc529138798)

[30 New Problems 36](#_Toc529138799)

[30a Effects on the Current Environment 36](#_Toc529138800)

[30b Effects on the Installed Systems 36](#_Toc529138801)

[30c Potential User Problems 36](#_Toc529138802)

[30d Limitations in the Anticipated Implementation Environment That May Inhibit the New Product 36](#_Toc529138803)

[30e Follow-Up Problems 36](#_Toc529138804)

[31 Migration to the New Product 37](#_Toc529138805)

[31a Requirements for Migration to the New Product 37](#_Toc529138806)

[31b Data That Has to Be Modified or Translated for the New System 37](#_Toc529138807)

[32 Risks 37](#_Toc529138808)

[33 Costs 37](#_Toc529138809)

[34 Waiting Room 37](#_Toc529138810)

[35 Ideas for Solutions 37](#_Toc529138811)

[36 Project Retrospective 38](#_Toc529138812)

[V Glossary 38](#_Toc529138813)

[VI References / Bibliography 38](#_Toc529138814)

[VII Index 38](#_Toc529138815)

### ****List of Figures****

[Figure 1 - Context of the work of making reservations, orders, and payments 16](#_Toc525544242)

[Figure 2 – Use Case Diagram of Reserve Your Table System 17](#_Toc525544243)

### ****List of Tables****

[*Table 2 - Requirements - Acceptance Tests Correspondence* 31](#_Toc525544241)

# Project Description

*Short Version ( SV ): Section I of the document provides a clear detailed picture of the product to be produced, why it needs to be produced, who would use it, what they would do with it, and provides other important background information prior to developing detailed requirements or designs.*

## Project Overview

*SV: Provide a brief quick description of the project, generally no more than a paragraph or two. The reader should get a good idea of what the project is all about from this opening section.*

Due to the current situation in the world, people are no longer dinning in restaurants as much as they used too. It's hard for both customers and restaurants to estimate the number of people who will dine-in in a particular day. Only way to check if there is room to dine-in is by giving the restaurant a call, which won’t guarantee the customer a table. This can be frustrating when a group of friends wants to have dinner and there aren't enough seats for them all.

*Reserve Your Table* aims to help people who used to love to dine in restaurants by allowing them to reserve a table at some of their favorite restaurants. People will be able to browse their favorite restaurants and reserve a table for themselves or a group to dine-in at the available selected time frame. Having interactions with unknown people is a currently a big concern for most. It is important to have the fewest number of interactions with people during these times. *Reserve Your Table* also aims to make their users feel safer when dinning in by allowing them to order their food once they are seated in their table. *Reserve Your Table* will also give the user the ability, once they’re finished eating, to pay for their bill using the application.

## The Purpose of the Project

*SV: Describe WHY this project is being done, and what one hopes to achieve from it.*

### The User Business or Background of the Project Effort

*SV: Describe the client’s business, e.g. the newspaper publishing business or the firefighting business, to the extent that it is relevant for this project. Note the distinction between “business” and “work” as described in section I.3 below.*

### Goals of the Project

*SV: Describe* ***WHY*** *this project is being carried out, from the point of view of the client. Note that the goal should be to improve the life of the client in some way, not just the development of software. (The SW is a means to an end, not the goal. )*

The goal for *Reserve Your Table* is to make users who used to dine in feel more comfortable when they dine in. Reserve your table aims to give the user a more pleasant way to dine in during this pandemic. This will also help user's local restaurants who are not doing so good during this pandemic. *Reserve Your Table* will give all the restaurants the ability to join the list of restaurants in the application, which will give the restaurants exposure to new potential customers.

### Measurement

*SV: How will one know when the goals stated in I.2.2b have been met? What measurable result can we point to and say that the goal has been met?*

The goals are to make the customers of the restaurants feel comfortable and satisfied to make their reservations to dine-in in the restaurants. The first measurement of the goal is to take and analyze the feedback from the customers through an integrated rating feature added into the application and responses from the survey accompany with the rating feature. The second measurement is to calculate the number of active users of the application. The third measurement is to get the data about sales information from the restaurants. The success of the goals will be met based on high ratings, number of users and sales of restaurants.

## The Scope of the Work

*SV: The “work” is a subset of the “business”, and describes the set of activities that will be addressed by the proposed product. For example, if the business is “university-level education”, then the work addressed by this project might be “the production and delivery of classroom lectures”. Obviously, the business of running a university encompasses a lot more than just classroom lectures, but this particular project will only concern itself with that particular aspect of the overall business.*

*A sentence or two here can briefly state what the “work” is.*

The *Reserve Your Table* application allow the user to track seats availability in the restaurants that registered to the system, reserve for spots and complete orders payment.

### The Current Situation

SV: Describe how the client is conducting the work now, without the proposed product. Note that the current situation may or may not involve computers.

In current time, the customers of the restaurants usually make phone calls to get information about the availability of the tables and reserve for their spots. In the situation that all the spots in a restaurant are already reserved, the customer will have to call another restaurant. This situation which is a bad experience for the customers can be solve using *Reserve Your Table* so that the customers can check the availability of spots in multiple restaurants without calling them. In addition, the payments of the orders usually happen at the receptionist’s counter of the restaurant. Due to the effect of the pandemic, human-human interactions should be limited. Therefore, the older payment method can be replaced by paying for orders directly from *Reserve Your Table* application*.*

### The Context of the Work

*SV: Define the boundary between what is included in “the work” and what is not. It also defines what external entities “the work” must interact with and what those interactions entail.* ***The following example diagram should be replaced with one appropriate to this project.***

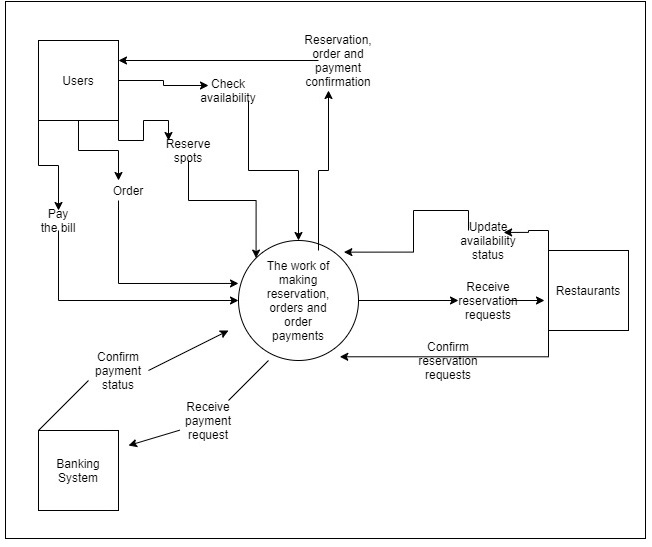


Figure 1. Context of the work of making reservations, orders, and payments.

### Work Partitioning

*SV: “The work” is often large and complex, with many different activities and concerns. One good way to break this down and organize it for analysis is to identify the different events to which the business must respond. A “business event” is an external stimulus which causes the business to take a series of actions in response.*

Your text goes here. *A table is recommended . . . (See full instructions for example.)*

|  |  |  |
| --- | --- | --- |
| **Event Name** | **Input/Output** | **Summary** |
| User attempts to login | User types in their email and password.  The system will verify this info by checking the database. If successful the system loads the users account and is prompt to the main UI. If failed, the system shows an error message. | Saving the users account in the database, allows the users can also view their reservations/orders/payment history as well as other account information. |
| User attempts to signup | User clicks on the register button.  The system will display a form for the user to fill out. If the form is filled correctly by the user, the system will store this user information in the database. And prompt the user to the main UI. If user failed to fill form correct, the system will display an error message. | New users need a way to create an account so all their information gets saved. |
| User finds and reserves a table | User selects a restaurant, table size, and time slot then clicks on confirm reservation button.  The system will send a message to the restaurant to confirm. The system will update the database with the user's reservation and display a confirmation message to the user. | If the user reserves a table, the user needs to know if their reservation was successfully placed. The system will update the user with the confirmation message. |
| User places an order in the restaurant | User adds items to an order, then clicks on submit order button.  The system will update info in the users account in the database, and then send a message to the restaurant with the order. | Again, the user needs to see if their order was placed, by getting a confirmation message from the system. |
| User attempts to pay for bill | User fills credit card form and clicks on submit payment.  The system processes the payment information. If successful the system will update the order balance in the database. If fail the system will display an error message. | The system needs to check the payment information in order to see if the payment went through or not. |

**Table 1: Work Partitioning**

### Competing Products

*SV:* ***IF*** *there are other existing products that the client could use instead of the proposed product, then they should be discussed here, along with the reasons why the proposed product is still needed / beneficial.*

OpenTable and Hostme are two existing products out there that the client could use instead. Hostme application is software that can be added to existing restaurant websites that only allow the client to only reserve a table. OpenTable also only allows the client to reserve a table. We believe that our product will be more beneficial to the client, since on top of being able to reserve a table, the client is also able to order their food and pay their bill with the application while they dine-in. Our application will also try to help small family-owned restaurants by allowing them to add their restaurant to the application for the clients to view.

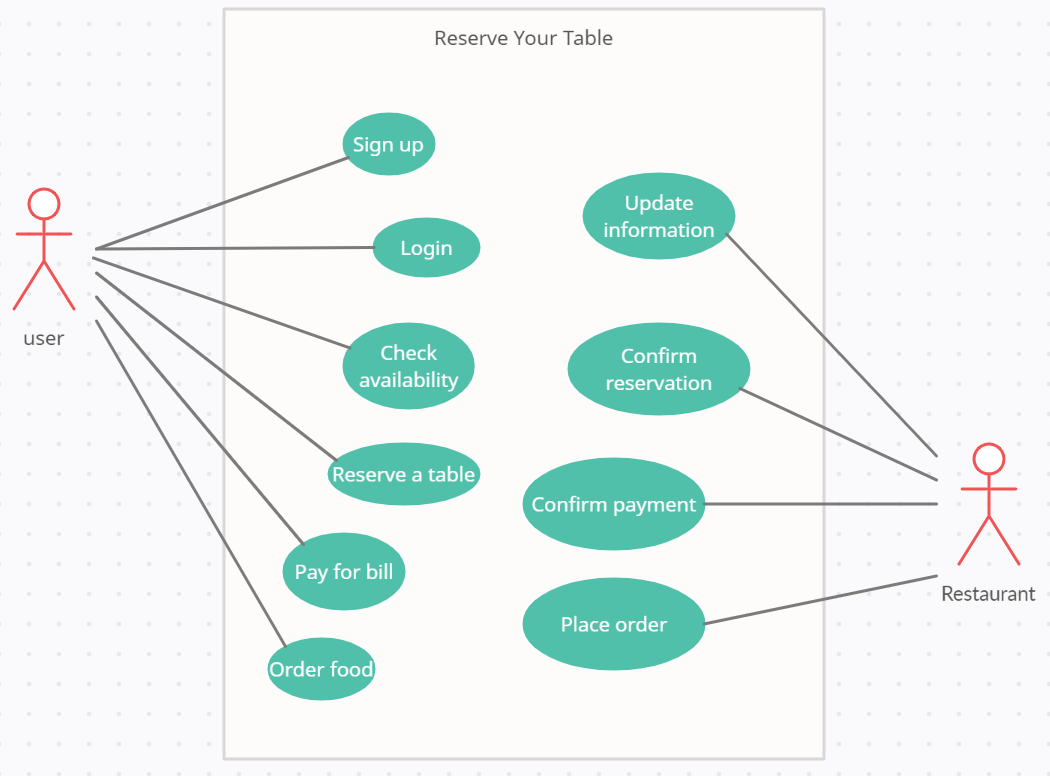
## The Scope of the Product

*SV: This section describes the proposed* ***product*** *as a set of short stories ( scenarios ) providing examples of how the product would be used in practice. This effectively documents what is and what is not included in the product, and who/what would interact with it in what ways. The opening paragraph briefly states what subset of “the work” is to be handled by the proposed product.*

Your text goes here . . .

### Scenario Diagram(s)

*SV: The scenario diagram acts as an illustrated list of the scenarios to be presented, showing the boundary of the system and what external “actors” are involved in each of the scenarios.*

Figure 2 - Use case diagram of Reserve Your Table System

### Product Scenario List

*SV: A table listing the scenarios by name, external actors involved, and possibly other information if relevant.*

**Sign Up:** The user can create an account to save their information. Their information, such as name and payments are saved in the cloud.

**Login:** User can login if they have an account created. The server sends the information like the restaurants local to the user for them to view.

**Check availability:** The user can check for the availability of the tables in the restaurant.

**Reserve a table:** Once the user reserves the table, the server sends to the restaurant a conformation message. After the restaurant sends the conformation to the server; the server sends a conformation email to the user and adds the information to the cloud so it can be stored in the user's reservation history.

**Order food:** The user can order the food directly from the application. The order will be saved on the server and the restaurants can check order accompanied with each reservation of their customers from the application.

**Pay for bill:** The user can pay for the bill directly from the application. The application is synchronized with their cards. When the user makes a payment, a payment request will be sent to the payment system of the restaurant. When the payment is confirmed, a message will be sent back to the user indicating that the payment has been completed.

**Update information:** The restaurants can update their information such as pictures, contact information and menus from the application. The application will send the updated data to the server.

**Confirm reservation:** The restaurants can confirm the reservation requests of customers who are either make reservations from Reserve Your Table application or

**Place order:**

**Confirm payment:**

### Individual Product Scenarios

*SV: This section contains the actual scenarios, the stories of the product being used.*

**Sign up:**

Mark is a university student. He has invited his friends to a restaurant near the place he lives. He has already downloaded and installed *Reserve Your Table* application to his phone from the application store. He wants to check the availability of the tables and reserve for the seats in the restaurant, so he opens the application to do so. The application loads and display a welcome screen, then the system shows a form that requires Mark to enter his username and password to login to the system. Since he does not have an account, he presses the sign-up button to register for an account. The system displays a form that requires him to enter his name, address, phone number, username, password, and information about his credit/debit card. After he has entered all the information, he pressed sign-up button and the system display a message notifying him that his account has been successfully created. The system then automatically switches the screen back to login screen where he can enter his username and password to login to the system.

**Login:**

After Mark has sign up for an account. The system goes back to the screen where he can log in using the username and password that he created. He enters his username and password then presses log in. The system verifies his information and display a message telling him that he has logged in successfully.

**Check availability:**

After Mark has logged in successfully. The system displays a screen showing the current reservation status. The system displays a message telling him that he currently has no reservation. He notices that there is a button shown “Find a restaurant”, so he clicks the button. The system displays a list of restaurants near the place he lives based on the address he has put in the system. He notices that there is a search box on top of the list, so he types in the name of the restaurant that he plans to go to with his friends. After he has typed in the name of the restaurant, the system displays only the name of the restaurant that matches what he typed, and that is the restaurant that he is looking for. He selects the name of the restaurant. The system displays a screen that shows a picture of the restaurant, and the information about the restaurant such as name of the restaurant, its address and the availability of the tables. The information shows that there are still 3 tables available right now in the restaurant.

**Reserve a table:**

After checking the availability of the tables. Mark notices that there is a button shown “Reserve a table” on the bottom of the screen. Mark selects the button. The system displays a form asking him to enter the information of the reservation. He enters the date and time when he will come with his friends, the number of people, and the number of tables. He then selects the confirm button to confirm the reservation. The system displays a message telling him that his table is successfully reserved. The system goes back to the screen that displays all of the reservations that he currently has. The screen displays the reservation that he has made on the list with the name of the restaurant, date and time of the reservation.

**Order food:**

After talking with his friends about the food that they will order at the restaurant, Mark open the application again. The system displays a screen asking him to enter his username and password. The system displays the list of his current reservations after he has logged in. Mark selects the reservation to the restaurant that he has made to come with his friends. The system displays a screen with the information of the reservation: the name of the restaurant, data and time, number of tables, number of people will come. There are three buttons on the bottom of the screen: “Order food”, “Arrived”, “Pay the bill”. Mark selects “Order food” button to order the food. The system shows a list of all the food available in the restaurant for him to choose from. He chooses the food that he has discussed with his friends and selects the number of portions for each type of food. Each type of food he added appears in a list on top of the screen with the number of portions on the side. He notices that he has added the wrong number for a type of food, so he removes it from the list and adds it again with the right number. After he has done choosing the food, he selects the confirm button to confirm the order. The system switches back to the screen that display his reservation. Now, the food that he has ordered appear in a list under the information of the reservation. Under the list of food that he has order, there is a line shows him the estimate time needed for the restaurant to serve all the food on the list (the time to cook all the food). The time for his food to be served is fifteen minutes.

**Pay for bill:**

As soon as Mark and his friends come to the restaurant. Mark opens the application again and navigates to reservation that is saved in his profile. The information on the reservation shows him that he has reserved table number 5. Therefore, he and his friends go to table number 5 and sit down. Mark then selects “Arrived” button to notify the restaurant’s staffs that he and his friends have arrived at the restaurants.

The staffs of the restaurant then start to prepare the food for Mark and his friends. Because the information on the list of his food said that he has to wait for fifteen minutes for the food to be served. Mark and his friends chat with each other while they are waiting for the food. After the food is served and they finish all the food. Mark selects the “Pay for bill” button. The system shows him the list of food, the number of each portion, the price for each type of food and the total price that he has to pay. After checking the payment information, Mark selects the “Confirm payment” button. The system then shows a message telling him to wait while the system is verifying the payment. After a few seconds, the system displays a message to notify Mark that his payment has been confirmed. The system displays a screen asking him to write a review and rate his experience in the restaurant. After he has done with the survey and submitted it, the system displays a message telling him that receipt of his payment has been saved to his profile.

## Stakeholders

*SV: Stakeholders include all persons or entities that have an interest in the proposed product or its development, either directly or indirectly.*

### The Client

*SV: The client pays up front for the product to be developed, and provides guidance or other input for its development. Some projects do not have an external client, in which case the developing organization acts as the client.*

The clients for the application will be organizations or companies that want develop a platform to allow the customers to register their restaurants to the system.

### The Customer

*SV: The customer is the person or entity who will buy the product after it has been completed. Some projects do not have an external customer, if they are to be used in-house or for the client’s use only.*

The customers of the application are the owners of restaurants who want to use the application to manage their reservation and payment processes.

### Hands-On Users of the Product

*SV: These are the people who will actually use the product in practice, and who may be separate from the customer or client. For example, educational software may be purchased by the school system ( customer ) and used by students ( hands-on users. )*

The hands-On users are the customers of the restaurants:

* + Username/category: restaurants’ customers
  + User role: the users use the application to make reservations and payment
  + Subject matter experience:
  + Technological experience: the users have the experience of using devices such as phones and computers in daily basis.
  + Other user characteristics:

### Maintenance Users and Service Technicians

*SV: Describe users that will install, maintain, update, and otherwise service the product as needed. May not apply to all projects.*

The maintenance users will be the development team of the application. They will maintain the application, fix the bugs reported from the users, and enhance the security of the application.

### Other Stakeholders

*SV: This section is a catch-all for all other stakeholders not previously mentioned. Note that some stakeholders may be negatively impacted by the proposed project, for example if their work duties change or are eliminated. Some may even object to the project all together.*

Your text goes here . . .

### User Participation

*SV: To what extent can we expect users to participate during the development of the product?*

User participation will be very important during the development. We will need feedback from beta testing users to inform about what features they like, and what we need to improve on. Users will be given access to different versions of the product, which will be improving the previous version according to their feedback. This will allow us to give the user the best experience we can.

### Priorities Assigned to Users

*SV: To the extent that some users are more important to the project than others, the relative priorities should be identified here.*

* + Primary users: Users who have created an account, they are able to browse restaurants around their location, reserve tables, order food, and pay.
  + Secondary users: Users who don’t currently have an account created. Only allowed to browse restaurants.

## Mandated Constraints

*SV: Mandated constraints are requirements that are set in stone by the client before the project is really even started, and before the full set of requirements are determined. Note that not all of these sections will apply to every project, and that some constraints could be placed equally well in more than one section ( but should not be duplicated. ).*

### Solution Constraints

*SV: These are general constraints on the product to be developed or the manner in which it is to be developed that are not covered elsewhere.*

* + - 1. The product should have separate application for the customer and the restaurants. The user and the restaurant will be doing different tasks so it is necessary that they each have their own application. The application for the customers should be a smartphone application which will be downloaded by using the android Play store app or Apple app store. The application for the restaurant should be a tablet application that will not be available publicly, only offered to registered restaurants.
      2. The customers application should be able to run on Android smartphones running Android 8.0 or higher, or on IOS smartphones running IOS 11 or higher. It's necessary to provide the application to android and apple users running these operating system versions because it will improve the user experience.
      3. The restaurants application should be able to run on any iPad with iPadOS 13 or higher. It must also be able to run on Android tablets running android OS version 10 or higher. A tablet has more screen room for the GUI, so it will allow to show more orders or reservations to the restaurants staff.
      4. The product should use *Stripe* as the payment infrastructure. This will allow the customer to pay through the application using one of the multiple payment options offered by *Stripe* such as Apple pay or Google pay*.*

### Implementation Environment of the Current System

*SV: This section deals with the physical and technical environment in which the proposed product will operate, such as hardware, operating system, and communications issues.*

When the application is installed on Android or IOS devices. If the client has an account, they should be able to sign in in either Android or IOS.

The application for the restaurants should be installed on an Android or IOS tablet. This software is only issued to restaurants that are in the list of restaurants on Reserve Your Table.

### Partner or Collaborative Applications

*SV: This section documents external applications with which this product must be compatible, such as the ability to read and write Microsoft Excel format data files.*

The application for the restaurants must have the ability to read and write data files in Microsoft Excel format so that the data about the menus of the restaurants can be loaded and modified directly from the application.

### Off-the-Shelf Software

*SV: This section describes commercial off-the-shelf ( COTS ) software that* ***MUST*** *be included in the final product.*

There is no COTS software needed. The customer will be able to install the application on their smartphone through the android app store or Apple app store. The restaurants are only required to download the application sent to them.

### Anticipated Workplace Environment

*SV: This section deals with human factors regarding the environment in which the product will be used, such as noisy environments or mobile applications.*

The application is not affected by the environment.

### Schedule Constraints

*SV:* ***When*** *things must be done, or when they may be most/least beneficial.*

### Budget Constraints

*SV: Limitations on the funds and other resources available for this project.*

Your text goes here . . .

## Naming Conventions and Definitions

*SV: Define terminology to avoid miscommunications or misunderstandings.*

### Definitions of Key Terms

*SV: Define* ***words*** *that may have special or multiple meanings.*

Bug: A bug can be an insect but, in this case, we are referring to errors that might occur in the application.

Synchronized: application will work together with the cards at the same time

Human-human: interactions between people in physical setting

### UML and Other Notation Used in This Document

*SV: Define* ***symbols****,* ***diagrams****, and other* ***notations*** *used. May refer to a standard reference, such as “UML Distilled” by Fowler. ( Include in bibliography. )*

### Data Dictionary for Any Included Models

*SV: Define* ***data structures*** *and* ***data properties*** *relative to this project, such as the contents of an employee record or the fact that student GPA ranges from 0.0 to 4.0 corresponding to letter grades of F to A. Data file formats may be referenced to documented standards, such as jpg or pdf.*

Restaurant's data:

User’s Information:

Ratings from 1 to 5

## Relevant Facts and Assumptions

### Facts

*SV: Factual information relevant to the project, such as census data.*

### Assumptions

*SV: Assumptions relevant to the project, such as the availability of necessary resources or abilities of the users.*

Your text goes here . . .

# Requirements

*SV: Sections 9 and 10 deal with functional requirements. Sections 11 to 20 are a very thorough list of possible non-functional requirements, not all of which apply to every project. You should think carefully about each of these, form requirements if applicable, or write “Not Applicable” otherwise. See section 10 for the format of individual requirements. Section 21 documents the acceptance tests planned to verify the requirements – See that section for further details, and be aware that every requirement needs at least one verifying acceptance test ( though some tests may verify more than one requirement. )*

## Product Use Cases

*SV: Product Use Cases are very similar to Product Scenarios, but in more formal detail. They serve as a first step towards developing functional requirements, and can aid in organizing requirements according to the use case(s) from which they were developed. See the CS 440 web site for a sample use-case form, with instructions.*

### Use Case Diagrams

*SV: Use case diagrams list the use cases developed for a system, mark the boundary of what is internal or external to the system to be developed, and indicate which external entities ( actors ) are associated with each use case.*

*Examples*

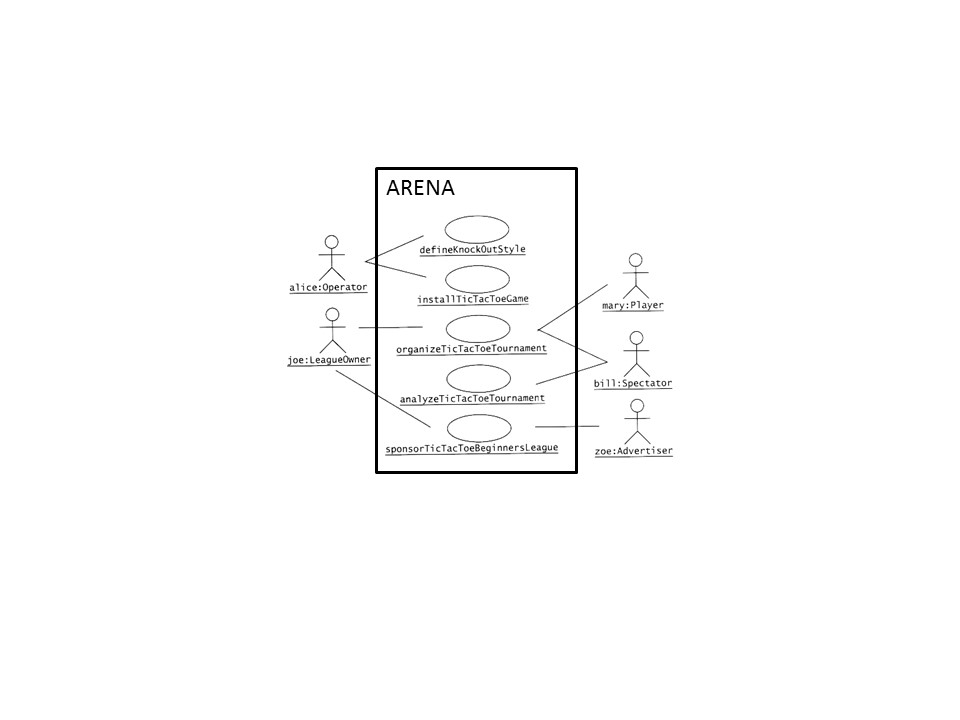


Figure 3 - Sample Use Case Diagram from Bruegge & DuToit ( modified )

**

Figure 4 - Sample Use Case Diagram from Robertson and Robertson

### Product Use Case List

*SV: A list ( table ) of use cases is an alternative to the use case diagram, particularly when there are many use cases. There may be additional information in the table not found in the diagram, such as cross referencing to other sections or materials.*

### Individual Product Use Cases

*SV: The following example was copied from “useCaseFormWithInstructions.docx”, available on the CS 440 web site. ( There is also a blank version available. )*

|  |
| --- |
| Use case ID: Name:  pre-conditions:  post-conditions:  Initiated by:  Triggering Event:  Additional Actors: |
| Sequence of Events:  Initiating event or action should be step 1, taken by initiating actor.   1. System response follows, indented right.   All external action steps are aligned with step 1. ( "stimulus" style )   1. All system responses are indented right, aligned with step 2. ( "response" style )   All steps should be expressed in the active voice, clearly indicating **who** performs each action   1. The sequence of events should show a back-and-forth stimulus-response relationship. |
| Alternatives: These would be normal and expected variations from the base case.  Exceptions: These would be unusual variations from the base case, often caused by problems. |

* *For all of the above, list as NA if not applicable.*
* *The following may be added if relevant, or omitted otherwise:*
  + *related use cases or scenarios*
  + *associated tests, systems, classes, etc.*
  + *revision history*
  + *references to other documents*
  + *author(s) / originator( s )*
  + *notes*
* *Alternatives and Exceptions may be listed either as separate use cases or as notes to a base case, depending on their significance and similarity.*
* *For regularly occurring periodic events, "time" can be listed as the initiating actor.*

## Functional Requirements

*SV: Each requirement listed needs to have a unique identifier, a short name, a one- or two-sentence description, a rationale, a fit criteria, and reference to one or more acceptance tests to be used to confirm the completion of this particular requirement. The acceptance tests themselves are documented in section 0- See that section for further details. It is recommended to number the requirements according to their type, such as F-4 for the fourth functional requirement or U-2 for the second usability requirement. Functional requirements specifically deal with the functionality the system must have, and are generally derived directly from the steps the system takes during use cases.*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

## Data Requirements

*SV: Data requirements deal with requirements that are somehow related to data, such as the definition of what is included in a “student record” or the acceptable form of an e-mail address or allowable range of certain data items.*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

## Performance Requirements

### Speed and Latency Requirements

*SV: Requirements specifying how fast ( or slow ) the product must operate or how much lag is allowable between stimulus and either initial response or task completion. Other timing-related requirements could go in this section.*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

### Precision or Accuracy Requirements

*SV: Self-explanatory. How accurate or precise must the system be.*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

### Capacity Requirements

*SV: Requirements regarding the largest “thing” the system must be able to handle, or perhaps how many things it can handle ( at once. ) Note: Requirements regarding how many things it can handle in a given time period would be a speed requirement, covered in section 12a above.*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

## Dependability Requirements

### Reliability Requirements

*SV: Reliability relates to how frequently the system fails, ( either by shutting down or by delivering erroneous results ), and the consequences of those failures. These requirements may also address the conditions under which it is allowed to fail ( or not. ), See also availability and robustness in the following sections.*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

### Availability Requirements

*SV: Availability addresses the amount of time the system is running and available for use. It is affected by how often the system goes down ( reliability ), but also by the time required to bring the system back up again, the availability lost due to regularly scheduled maintenance down times, and the ability of the system to offer at least partial functionality in the face of failures or resource shortages. See also reliability and robustness.*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

### Robustness or Fault-Tolerance Requirements

*SV: This section deals with the system’s ability to provide at least partial functionality in the face of failures or resource shortages, such as operating in offline mode when network connectivity is unavailable. See also reliability and availability.*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

### Safety-Critical Requirements

*SV: These requirements address potential harm to health, safety, or property, and may refer to relevant standards such as OSHA compliance.*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

## Maintainability and Supportability Requirements

### Maintenance Requirements

*SV: This section deals with the ease with which the system can be maintained, and possibly who will perform system maintenance and under what conditions. The ease of evolving the system into future versions may also be addressed here, or in a separate section ( not included in this template ) if that is a major concern.*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

### Supportability Requirements

*SV: What ongoing support is to be provided, e.g. through a help desk. See also training requirements in section 16g below.*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

### Adaptability Requirements

*SV: Description of other platforms or environments to which the product must be ported.*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

### Scalability or Extensibility Requirements

*SV: The ease of expanding the system to a larger capacity as the business grows.*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

### Longevity Requirements

*SV: This specifies the expected lifetime of the product.*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

## Security Requirements

*SV: Security requirements address who is allowed what type of access to the system, and what areas require special protection or diligence. In practice security requirements must often be written by security experts, and may refer to standards.*

### Access Requirements

*SV: These requirements address who has access to what ( data or functionality ) and under what conditions or restrictions.*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

### Integrity Requirements

*SV: These requirements address the protection of data(bases) from intentional or accidental corruption, loss, or theft.*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

### Privacy Requirements

*SV: These requirements address data that must remain confidential, such as medical records or other personally identifiable data. Laws often apply. (See also section 20.)*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

### Audit Requirements

*SV: This section applies when a system must provide support for transaction auditing, such as some financial or medical systems.*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

### Immunity Requirements

*SV: This section addresses the system’s ability to resist viruses, worms, Trojan Horses, etc.*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

## Usability and Humanity Requirements

*SV: This section is concerned with requirements that make the product usable and ergonomically acceptable to its hands-on users.*

### Ease of Use Requirements

*SV: This section addresses the ease with which the intended audience can use the system properly, and conversely the difficulty with which they can use it improperly.*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

### Personalization and Internationalization Requirements

*SV: This section addresses the ease with which the system can be configured for personal preferences, and for things such as language, currency, units, symbols, etc.*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

### Learning Requirements

*SV: Requirements related to how easy it is for the intended audience to learn to use the product.*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

### Understandability and Politeness Requirements

*SV: These requirements relate to how intuitively the intended audience understands what the program does, what its messages mean, and how to use it. Definitely related to ease of use, ( section 16a ), but more specifically addressing comprehension of the program output, instructions, and other messages.*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

### Accessibility Requirements

*SV: Requirements related to use of the product by individuals with disabilities.*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

### User Documentation Requirements

*SV: List of the user documentation to be supplied as part of the product.*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

### Training Requirements

*SV: A description of the training needed by users of the product.*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

## Look and Feel Requirements

### Appearance Requirements

*SV: These requirements address things such as the colors, fonts, and logos used, often to reflect corporate branding or similarity to related products. See also style in the next section.*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

### Style Requirements

*SV: Style requirements address the impression the product makes upon users, such as professionalism for a tax accounting package, friendliness for a children’s game, or how “cool” it is for a teenage audience. Product packaging may also be addressed here, and/or appearance in the previous section.*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

## Operational and Environmental Requirements

### Expected Physical Environment

*SV: These requirements relate to the physical environment in which the product will operate.*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

### Requirements for Interfacing with Adjacent Systems

*SV: This section describes the requirements to interface with partner applications and/or devices that the product needs to successfully operate.*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

### Productization Requirements

*SV: Requirements related to the distribution and/or installation of the product.*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

### Release Requirements

*SV: Specification of the intended release cycle for the product and the form that the release shall take.*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

## Cultural and Political Requirements

### Cultural Requirements

*SV: This section contains requirements that are specific to the sociological factors that affect the acceptability of the product. If you are developing a product for foreign markets, then these requirements are particularly relevant. Bear in mind that “cultural groups” may also apply to population subgroups such as teenagers, the elderly, or ironworkers.*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

### Political Requirements

*SV: Requirements included strictly to make “the boss” happy, either internally to the development company, or internally to the client company, or possibly an external third party.*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

## Legal Requirements

### Compliance Requirements

*SV: A statement specifying the legal requirements for this system, often referring to relevant laws and/or requiring approval by the legal department.*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

### Standards Requirements

*SV: These requirements specify documented standards to which the product must conform, as opposed to legal regulations.*

**ID# - Name**

**Description:** Your description here . . .

**Rationale:** Your rationale here . . .

**Fit Criterion:** Your fit criteria here . . .

**Acceptance Tests:** List ID# and/or names here . . .

## Requirements Acceptance Tests

*SV: Every requirement must have one or more acceptance tests associated with it, to confirm that the requirement has been met. At this point these tests are not yet completely specified – A one- or two-sentence description of each test will suffice. Note that some tests may verify more than one requirement, and that some requirements may require multiple tests for their confirmation.*

### Requirements – Test Correspondence Summary

*SV: The following sample table is available from the CS 440 web site as “Sample Requirement Test Correspondence Table.xlsx” It is recommended that you work with the table in Excel, and then drag it into the document when it is completed. Depending on the number of requirements and/or tests included, it may be necessary to use multiple tables, and/or use landscape mode. Every row and every column of the table should include at least one X. Below the table list the ID #, name, and short description of each individual acceptance test.*

**

*Table 1 - Requirements - Acceptance Tests Correspondence*

### Acceptance Test Descriptions

*SV: Provide a brief description of each acceptance test. Detailed test specifications will appear in a separate document, which may be referenced here when available.*

**ID # - Name**

**Description:** Your description here . . .

# Design

## Design Goals

*SV: Identify the important design goals that are to be optimized in the proposed design.*

Your text goes here . . .

## Current System Design

*SV:* ***IF*** *the proposed new system is to replace an existing system, then the current system should be described here. Otherwise insert a brief statement that there is no pre-existing system.*

Your text goes here . . .

## Proposed System Design

*This section will make heavy use of class diagrams, and also sequence and deployment diagrams where noted. However don’t overlook finite state, activity, communication, or other diagram types as needed for effective communication.*

### Initial System Analysis and Class Identification

*SV: Perform grammatical and similar analyses to identify the most import and obviously needed classes, and to organize them into an initial class structure. An initial class diagram is appropriate, containing few if any internal details.*

Your text goes here . . .

### Dynamic Modelling of Use-Cases

*SV: Insert sequence diagrams of ( at least the most important ) use-cases, as a means of identifying other needed classes.*

Your text goes here . . .

### Proposed System Architecture

*SV: Identify the Software Architecture to be applied to this project, such as Client-Server, Repository, MVC, etc., along with justification for the choice.*

Your text goes here . . .

### Initial Subsystem Decomposition

*SV: A slightly more detailed class diagram, showing the classes identified in sections 24a, 24b, and 0 above, partitioned into subsystems. For each subsystem provide a brief description of the subsystem, including its key responsibilities. There should still be few if any internal details.*

Your text goes here . . .

## Additional Design Considerations

*SV: The sections listed here do not need to be presented in the order given, and may not all be relevant for any particular project. Those that are relevant can help identify additional classes that are needed as a result.*

### Hardware / Software Mapping

*SV: This is particularly important for distributed systems, such as those employing a client-server architecture. Use a deployment diagram to indicate which subsystems are mapped onto which piece(s) of hardware, and what communication subsystems need to be added to the system as a result.*

Your text goes here . . .

### Persistent Data Management

*SV: Document the classes and perhaps subsystems necessary to store persistent data when the system shuts down, and to restore that data when the system starts back up again.*

*Reiterate key data structures and information as necessary for the understanding of this design phase. Refer the reader back to the data dictionary in section* ***Error! Reference source not found.*** *to avoid undue repetition, while reviewing only the most relevant items here.*

Your text goes here . . .

### Access Control and Security

*SV: Identify the access control and security concerns for this system, and the new classes and/or subsystems that must be added to handle those concerns.*

Your text goes here . . .

### Global Software Control

*SV: Identify the global software control concerns for this system, and the new classes and/or subsystems that must be added to handle those concerns.*

Your text goes here . . .

### Boundary Conditions

*SV: Identify the boundary condition concerns for this system, and the new classes and/or subsystems that must be added to handle those concerns. In particular consider startup, shutdown ( normal or abnormal ), and the creation and/or maintenance of any configuration files, databases, or similar supporting data files.*

Your text goes here . . .

### User Interface

*SV: Include a preliminary user interface design here, possibly as a rough sketch or other mockup, in order to identify additional classes needed to implement the interface.*

Your text goes here . . .

### Application of Design Patterns

*SV: Any design patterns applied as a result of previous sections should have been addressed there, and identified as such at the time. Use this section to document only the additional design patterns that were not previously covered elsewhere. ( If any. )*

Your text goes here . . .

## Final System Design

*SV: Include here the final version of the overall system design, incorporating all the subsystems and classes added as a result of additional design considerations. Multiple diagrams may be needed, possibly starting with an overall package diagram showing all the different subsystems and the ( important ) classes contained within each one. Still not a lot of internal details.*

Your text goes here . . .

## Object Design

*This section documents the internal details of each class, to the extent that they can be designed at this time. Included should be the class interfaces ( public method signatures and responsibilities ) and constraints. It is probably best to break this section up into subsections corresponding to subsystems as documented above, and/or by ( Java ) packages if those are designed. It may also be appropriate to address additional design pattern considerations here, but not to the point of being redundant of previous documentation.*

*Certain methods, such as simple getters, setters, and constructors are not always documented, unless there is something special about them such as in the Singleton or Factory Method design patterns.*

### Packages

*SV: If the design involves assigning classes to packages ( .e.g Java packages ), then the packages to be created should be documented here.*

Your text goes here . . .

### Subsystem I

Your text goes here . . .

### Subsystem II

Your text goes here . . .

### etc.

Your text goes here . . .

# Project Issues

## Open Issues

*SV: Issues that have been raised and do not yet have a conclusion.*

Your text goes here . . .

## Off-the-Shelf Solutions

*SV: Discussion of products or components currently available that could either be incorporated into the new solution or simply used instead of developing ( parts of ) the new solution.  The distinction between sections 35 a, b, and c is subtle, and not very important.*

Your text goes here . . .

### Ready-Made Products

*SV: Products available for purchase that could be used either as part of a solution or instead of ( a part of ) a solution.*

Your text goes here . . .

### Reusable Components

*SV: Similar to 35a, but for components such as libraries or toolkits instead of fully blown products.*

Your text goes here . . .

### Products That Can Be Copied

*SV: Products that could legally be copied would typically be past projects developed by the same development group, provided there were no restrictions that would prevent their reuse.*

Your text goes here . . .

## New Problems

*SV: The proposed new system certainly has its benefits, but it could also raise new problems.  It is a good idea to identify any such potential problems early on, rather than being surprised by them later.*

### Effects on the Current Environment

*SV: Could the new system have any adverse effects on the working environment, e.g. the way people do their jobs?*

Your text goes here . . .

### Effects on the Installed Systems

*SV: Could the new system have any adverse effects on other hardware or software systems?*

Your text goes here . . .

### Potential User Problems

*SV: Could the new system have any adverse effects on the users of the software? Could users possibly have a negative response to the new system?*

Your text goes here . . .

### Limitations in the Anticipated Implementation Environment That May Inhibit the New Product

*SV: Are there any ( physical ) limitations in the expected environment that could inhibit the proposed product?  ( e.g. weather, electrical interference, radiation, lack of reliable power, etc. )*

Your text goes here . . .

### Follow-Up Problems

*SV: Basically any other possible problems that could occur.*

Your text goes here . . .

## Migration to the New Product

*SV: This section only applies when there is an existing system that is being replaced by a new system, particularly when data must be preserved and possibly translated / reformatted.  Otherwise just write "Not Applicable" under section 38 and remove sections 38a and 38b.*

### Requirements for Migration to the New Product

*SV: These are a list of requirements relevant to the migration procedures.  For example a requirement that the two systems be run in parallel for a time until the client is satisfied with the new system and the users know how to use it.*

Your text goes here . . .

### Data That Has to Be Modified or Translated for the New System

*SV: This section specifically addresses****data****that must be preserved and/or translated / reformatted during the migration process.*

Your text goes here . . .

## Risks

*SV: Consideration of the potential risks that could cause the project to fail / underperform.*

Your text goes here . . .

## Costs

*SV: An estimate of what it will cost to complete this project.  Think not only in terms of dollars, but also time, resources, lost opportunities, etc.*

Your text goes here . . .

## Waiting Room

*SV: This is a place to record ideas or wishes that will not be included in the current release of the product, but which might be worth reconsidering at a later date.*

Your text goes here . . .

## Ideas for Solutions

*SV: When developing requirements only, it is not the role of the business analyst to dictate the implementation of the solution.  However they can pass along any ideas they have here as suggestions to the developers.  For CS 440 this report includes system and object design, so this section would make suggestions for implementation and testing that would come after design, such as the use of a particular language, IDE, library, or other tools.*

Your text goes here . . .

## Project Retrospective

*SV: At the conclusion of the ( CS 440 ) project, reflect back on what worked well and what didn't, and how the process could be improved in the future.*

Your text goes here . . .

# Glossary

*SV: The glossary is a more complete and inclusive dictionary of defined terms than that found in section I.7.a, the latter of which only covered the most important key terms needed to understand the report.*

Your text goes here . . .

# References / Bibliography

*This section describes the documents and other sources from which information was gathered. This sample bibliography was generated using the “Insert Citation” and “Bibliography” buttons in the “Citations & Bibliography” section under the “References” tab of MS Word. Creating new citations will not update this list unless you click on it and select “Update Field”. You may need to reset the style for this paragraph to “normal” after updating.*

|  |  |
| --- | --- |
| [1] | Robertson and Robertson, Mastering the Requirements Process. |
| [2] | A. Silberschatz, P. B. Galvin and G. Gagne, Operating System Concepts, Ninth ed., Wiley, 2013. |
| [3] | J. Bell, "Underwater Archaeological Survey Report Template: A Sample Document for Generating Consistent Professional Reports," Underwater Archaeological Society of Chicago, Chicago, 2012. |
| [4] | M. Fowler, UML Distilled, Third Edition, Boston: Pearson Education, 2004. |

# Index

*This section provides an index to the report. The sample below was generated using the “Mark Entry” and “Insert Index” items from the “Index” section on the “References” tab, and can be automatically updated by right clicking on the table below and selecting “Update Field”. To remove marked entries from the document, toggle the display of hidden paragraph marks ( the paragraph button on the “Home” tab ), and remove the tags shown with XE in { curly braces. }*

Design 61, 63

Requirements 35, 51, 58

Test 64, 65