**Group 4**

Jeremy Moses

Andrea O’Brien

Chris Womack

Eric Whatley

Yi Chen

**Due 11/12/2013**

CS310 – Software Engineering I

Lot Reservation System

**Report 4**

Table of Content

Description of Business .…………..…………………..………………………………………… 3  
Business Need ……………………..……………………………………………………….…… 4  
Project Overview ……………….…………....…………………………………………….……. 5  
Scope of the System ………………………………………………………………….………… 6  
 Functional Requirements ………………………………………………………..……… 8  
 Non-Functional Requirements …………………………………………………….….… 9  
Stakeholders ……………………………………………………………………………………. 10  
Team Configuration ……………………………………………………………………….…… 11  
Feasibility Analysis ………………………………………………………………………….…. 14  
Work Plan ……………………………………………………………………………………… 20  
 Work Breakdown Schedule …………………………………………………….……… 20  
 Use Case Points Analysis……………………………………………………………….. 25 Gantt Chart ………………………………………………………………………….….. 27  
Modeling the System…………………………………………………………………………… 29  
 Use Cases ……………………………………………………..………………………... 29  
 Use Case Diagram ………………...……………………...…………………….. 29  
 Use Case Description ………………..…………………………………………. 30  
 CRC Cards ………………………………………..……………..………………….….. 36  
 Class Diagram .…………………………………………………………………….….. 43  
 Activity Diagrams …………..………………………………………………………….. 44  
 Sequence and Communication Diagrams ……………..……………………..………… 48  
 State Machine Diagrams ………………..……………..……………………..………… 54  
User Interface Design …………………………..……………..……………………..………… 55  
 Description of Interface Standards ……..……………..……………………..………… 55  
 Windows Navigation Diagram ………..……………..……………………..………… 57  
 Screenshots/Prototypes ………………..……………..……………………..………… 59  
 Design Choices Justifications ….……..……………..……………………..………… 61  
Appendix ……………………………………………………...……………………..…………. 62

**Description of the Business**

* **Product of Services Offered**

Mountain Top Flea Market offers a rich environment where people can come together to show off and sell their goods, buy other goods, and have a nice, relaxing day. The market has hundreds of vendors that sell clothes, toys, tools, electronics, plants, produce, and animals.

* **Size of the Business**

Mountain Top Flea Market has hundreds of vendors, and many more buyers. People come from all over the state, and sometimes, the country, to set up or shop at Mountain Top.

* **Brief Description of As-Is System**

The As-Is system is pen and paper based. The supervisors of the flea market record all of the information from the vendor using paper. They record name, date, lot number, cost, amount paid, change given, time arrived, and many other things. Using this system leads to reservation discrepancies and increases the chance for error.

* **Brick and Mortar**

This business is brick and mortar. Mountain Top Flea Market is made up of hundreds of booths of varying sizes, both covered and uncovered. It is based outside with a dirt and rocky floor throughout. The vendors can rent a booth with a floor.

* **Customer Base / Markets Served**

The customer base for Mountain Top Flea Market is relatively large for a flea market, and it’s still growing. There are many vendors that come to set up at Mountain Top. The people that walk around are not directly our customers, but the more buyers there are, the more vendors will be encouraged to come back.

* **Competition for the Business**

The competition for Mountain Top Flea Market is other flea markets such as Attalla Trade Day. Other competitions could include yard sales, and any other place where goods are sold. We strive to make our market as friendly and appealing to customers as possible so that we may minimize competition.

**Business Need**

It has become increasingly hard and time-consuming to keep up with the hundreds of rented spaces. Using paper records to keep up with reservation can lead to reservation discrepancies. The business needs an updated system that will allow them to easily and efficiently rent out spaces to the vendors. Additionally, by solving this need the business will now be able to reduce errors and save more trees by not wasting paper. This will also help to reduce the traffic problem because it will allow vendors to be checked-in at a faster rate.

**Project Overview**

Our company needs to improve the process of managing reservations in order to decrease wasted resources and increase revenue. There are many problems with the as-is system. One such problem is the reservation department is having difficulty assigning and keeping track of the lots that are still available for rent. Another problem is the customer service department is experiencing issues with determining which vendors still need to pay. We need to create a new system in order to help our company better handle these issues. The new system will be more accurate and cause fewer errors which will make the vendors happier and be willing to come back more often.

The new system will replace the manual system that we already use. The system will decrease the amount of discrepancies caused by the manual system. The new system will help reduce common errors, which will save the company money. It is a system that can be built within a time frame of six months. It will be utilized by the employees in the reservation department to map out, assign, and charge a vendor for each available lot. It will keep track of all the available lots on the land and will show which vendor is assigned to each reserved lot. The customer service department will be able to easily determine who needs to pay and lookup a customer to see if they reserved a space. For security purposes, the system will not be able to access the Internet. At this time, the map that is used to show the status of each lot cannot be altered by the users. In later versions we plan to incorporate the ability to track the history of customers, past events, and transactions.

The development team consists of people who have all the specialties needed to create the system. The benefits of the system will outweigh the costs to develop the system. In order to complete the system we will need several types of programmers. We would also need to have a server, client work stations, and printers. To complete the system an analyst team is also needed. The estimated time to complete the new system is six months.

**Scope of System**

* **Overview**

The new system will be used by the employees in the reservation department to map out, assign, and charge a vendor for each available lot. It will keep track of all the available lots on the land and will tell the market staff which vendor is assigned to each lot that is filled. The customer service department will be able to easily determine who to needs to pay and if a customer has reserved a lot or not. The system will not be able to access the Internet. The system will not allow the users to change or update the map of the flea market grounds. In later versions, the system will be able to track the history of customers, past events, and transactions. It will also allow the user to customize the layout of lots in the system.

* **Requirements Gathering Methodology**

In order to gather all requirements needed for the proposed system, we used the following requirement methodologies. We assigned a team to observe the manual process that was used at Mountain Top Flea Market. This allowed our team to understand the as-is system requirements so we can implement the requirements into an updated system. The team also performed a document analysis on the manual system to fully understand the process. While the assigned team was located at the Mountain Top Flea Market, they conducted interviews with the management and the employees that would use the new system. We also gave questionnaires to everyone we interviewed so that we could have even more data. To reduce time, the interview questions were already analyzed and agreed upon. The interviews provided information about requirements that the different departments would need the new system to fulfill. Finally, the team conducted several JAD meetings. This brought the team working on the system and the users of the system together and finalized the requirements needed for the system. Each method gathered all of the requirements needed for the new system. In addition, we decided that we should interview some of the long-term vendors to understand what they would like to be improved about the way reservations are done.

* **Overall System Objectives**

The main objective of the new system is to simplify the process of reserving lots at Mountain Top Flea Market. The system will transfer the manual system into an easily accessible software system. The system will map out, assign, and charge a vendor for each available lot. It will keep track of the available lots on the land and will tell the market staff which vendor is assigned to each lot that is filled.

* **Functional Requirements** 
  + **F1. Manage Lot Reservations:**
    - **F1.1** The system will allow the system user to make new reservations.
    - **F1.2** The system will allow the system user to make changes to customer’s reservations (change assigned lot location, change reservation date, ect.).
    - **F1.3** The system will allow the system user to cancel lot reservations.
    - **F1.4** The system will update reservation map whenever changes are made.
  + **F2. Access Reservation Map**
    - **F2.1** The system will allow the system users to review location map of reserved lots.
    - **F2.2** The system will allow the system users to print location map of reserved lots.
  + **F3. Charge Vendor for Reserved Lots**
    - **F3.1** The system will allow management to set separate charges per type of lot reservation.
    - **F3.2** The system will allow the management to update the prices for each lot type and extras.
* **Nonfunctional Requirements**
  + **NF1. Operational Requirements**
    - **NF1.1** The system will operate in a Windows environment.
    - **NF1.2** The system will connect to printers wirelessly.
    - **NF1.3** The system will automatically back up system changes at the end of each day.
  + **NF2. Performance Requirements**
    - **NF2.1** The system will store new lot in 3 seconds or less.
    - **NF2.2** The system will retrieve the daily lot reservation locations in 3 seconds or less.
    - **NF2.3** The system will update each lot reservation in 3 seconds or less.
  + **NF3. Security Requirements**
    - **NF3.1** Only Mountain Top Flea Market management can change the lot rates in the system.
    - **NF3.2** OnlyMountain Top Flea Market authorized employees can log into the system.
    - **NF3.3** Only Mountain Top Flea Market authorized employees can reserve lots.
  + **NF4. Cultural and Political Requirements**
    - **NF4.1**There are no special cultural and political requirements.

**Stakeholders**

* Business Owners need the ability to generate and print reports; the ability to add and remove authorized users; and the ability to change lot information and pricing.
* Managers\Employees need the ability to add, edit or cancel reservations; the ability to view, validate and print vendor information; the ability to view and print a lot available map; and the ability to generate and print invoices.
* Vendors need a more accurate and easier way to reserve a lot.

**Team Configuration**

* **Team Lead**
  + **Jerry Matters** 
    - Mr. Matters has been an SAP HCM practitioner for over 20 years, spending his early years with PwC Consulting, and IBM Global Business Services. He has also led the analysis, design **configuration**, and implementation of Enterprise Compensation Management (ECM) module for many of Coke’s major development plans.
    - Specialties: SAP Talent Management including Performance Management, Enterprise Compensation Management (ECM), Succession Planning, Career Development, and E-Recruiting; Employee Self Service, Manager Self Service, Self-Service Portals, Usability/User Experience.
* **Systems Analysis**
  + **Chris Womack**
    - He graduated from Montana State University with a B.S. in Accounting and Data Processing and got a MBA in Business and Operations Research from The University of Maryland.
    - He designed and implemented the new **system for Rocketeers Management Services**, overhauled their ineffective **system**, and orchestrated sweeping changes with the new management **system** in less than one year, saving 15% of their annual labor costs. He also defined a new product development methodology and drove the completion of the new World Hikers Club Card **system** in nine months, reducing the product development cycle by 35% and deliver new products ahead of plan for them.
* **Programmers**
  + **Eric Whatley**
    - He graduated from The University of Oregon with a B.S in Computer Science and got a Master’s in Computer Science from Oxford.
    - Specialties: **Java** Enterprise, **Java** EE, J2EE, Sencha Touch, Spring, Hibernate, HTML5, Ajax, Mobile, Tablet, Android, iOS, iPhone, iPad, Cloud Computing, Grails, Groovy, REST, Cloud Foundry, Heroku, Git, Github. **Java** Play, and many other languages.
  + **Jeremy Moses**
    - He graduated from Arizona State University with a B.S in Zoology, got a B.S in Information Technology and a M.B.A in Information Technology Management from Washburn University.
    - Specialties: CGEIT (Certified in the Governance of Enterprise IT), CISA (Certified Information Systems Auditor), CISSP (Certified Information Systems Security Professional), CIW v5 Professional (**Database** Design, Site Design, and JavaScript), MCP (Microsoft Certified Professional), Network+ (CompTIA ), CHPSE (Certified HIPAA Privacy Security Expert)
  + **Andrea O’Brien**
    - She graduated from University of California-Berkley with a B.S in Computer Science and M.S. in Computer Science from Carnegie Mellon.
    - Specialties: Java, HTML, Full life cycle development, Process Management, Software Development Life Cycle SDLC, Oracle, DB2, MySQL, GWT, HTML 5, jQuery, NetBeans Platform, CA SiteMinder, RedHat EL, Suse Linux, Centos, Mac OSX
  + **Yi Chen**
    - He graduated from University of British Columbia with a B.A in Computer Science.
    - Specialties: Java, Python, C++, SVN, CVS, Mercury Test Director, Bugzilla, JIRA, Mantis, Twiki, Ant, HTML, XML, SQL, jBASE, T24, shell and VB

**Feasibility Analysis**

* **Technical Feasibility Analysis**

The system technology is new for the Mountain Top Flea Market business employees since the original system that the business used is a manual paper system. The system technology is not new for the development team. Since our analyst that is working on the system is good at understanding what the system needs to meet all functionality requires of the user there should be very little revisions to the procedures. The amount of people working on the system consists of six people; the team leader: Jerry Matters, the system analysis: Chris Womack, and three programmers: Eric Whatley, Jeremy Moses, and Andrea O’Brien. Since the team working on the system only consists of six people, fewer problems will arise. Typically, the more people involved in the development process of a system, the greater the risk. The estimated time it will take to complete the system is six months. Since the system does not contain a complex setup, the team can take an extra amount of time understanding all of the system requirements. Even though the system is new to the users and business members, the system will be compatible and easy to integrate into the Mountain Top business. Currently, the business does not have an existing technology. Since the system’s purpose is to ease the process of reserving lots, it will be easy to integrate the system. There are minor risks in creating the system, but since the system is specifically made for Mountain Top Flea Market, and it is a relatively small project, the risks of building the system are very low. The overall project size is relatively small because the development team is a small group, and the system is not too large. Therefore, there is a very small risk to build the system, and the development team should be able to build it.

* **Organizational Feasibility Analysis**

The Lot Reservation System is designed specifically for Mountain Top Flea Market. It is up to the stakeholders of the system to determine the success of the system. One of the main stakeholders for this system is the system champion. The champion of this system is the owner of Mountain Top Flea Market. Kyle Hampton is the current owner of Mountain Top Flea Market, and he is the person who proposed the idea of making the Mountain Top Interface system. The project champion will work with the management team at Mountain Top Flea Market to promote the new system. A series of presentations will be conducted with the employees of Mountain Top Flea market about the objectives of the system. The main objective of the system is to ease the process of reserving lots at Mountain Top Flea Market. Since the main objective is also a benefit to the organization, the presentations will be appealing towards the users. Jerry Matters is the team leader of the system and he will work closely with the project champion to promote the system to the users. The success of the system will be determined by the employees of Mountain Top Flea Market (the users). They will ultimately determine whether the system is useful to their business or not. To ensure that the users are satisfied with the system, the development team will work offer to train them. The development team will ask for feedback from the employees to ensure the system meets all users’ requirements. This will increase the systems alignment with Mountain Top Flea Market. The strategic work of the project champion and the project management will allow the users of the system to be satisfied and willing to use the system. The process involved in the organizational feasibility analysis will allow the users to accept and be satisfied with the new system.

* **Economic Feasibility Analysis**

The benefits include increased sales, reduced labor and reduced materials to total $110,000 in the first year which increases by $500 per year. The increased sales value is based on gaining one more vendor each week for the first year and one more vendor each year after the first year. The increased sales for each week for the first year would be shown in the equation: the sum of 1+2+3+…+52 times $10. Since it is a bespoke system we do not need and any software licenses for our system. Thus, the development costs only include development labor and a server to total $250,000. Operational costs include hardware and operational labor to be $40,500 a year. At the end of five years we estimate to have a profit of $66,921.85 with a return on investment of 16.19%. The break-even point is in 3.81 years.

* **Feasibility Analysis Summary**

The feasibility analysis is an important factor in the completion of the Mountain Top Interface system. The economic feasibility analysis shows that in a little less than four years, the system will reach the break-even point, from the saving earned by reduced cost of labor and materials. The technical feasibility analysis will ensure the development team, the management team, and the major stakeholders that the system can be built. Since the system is considered to be a simple system, the time frame to complete the system is relatively small. Essentially, the risks to make the system are very low. The organizational feasibility analysis ensures that the users will ultimately use the system. The purpose of making the system to ease the reservation system the Mountain Top Flea Market already uses. Since the system is specifically designed for the business, the user will use the system because it will be easy to use and make their jobs must simpler. To ensure the users are satisfied with the final product, the team leader and the team champion will promote and work with the users of the system. The feasibility process will force the development team to consider the economic, technical, and organizational factors that will affect the outcome of the system. As the development process increases, the feasibility analysis will change upon major decisions made with the system.

**Economic Feasibility Analysis**

\*The present value data is found by using a five percent interest rate

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Benefits** |  |  |  | **2013** | **2014** | **2015** | **2016** | **2017** | **Total** |
| Increased sales | 15000 |  | Increased sales | 15,000.00 | 15,500.00 | 16,000.00 | 16,500.00 | 17,000.00 |  |
| Reduced labor | 90000 |  | Reduced labor | 90,000.00 | 90,000.00 | 90,000.00 | 90,000.00 | 90,000.00 |  |
| Reduced materials | 5000 |  | Reduced materials | 5,000.00 | 5,000.00 | 5,000.00 | 5,000.00 | 5,000.00 |  |
| **Total Benefits** | 110000 |  | **Total Benefits** | 110,000.00 | 110,500.00 | 111,000.00 | 111,500.00 | 112,000.00 |  |
|  |  |  | **PV of Benefits** | **104,761.90** | **100,226.76** | **95,885.97** | **91,731.33** | **87,754.93** | **480,360.89** |
| **Development Costs** |  |  | **PV of all Benefits** | **104,761.90** | **204,988.66** | **300,874.64** | **392,605.96** | **480,360.89** |  |
| Development labor | 200000 |  |  |  |  |  |  |  |  |
| Server | 50000 |  | Development labor | 200,000.00 | 0.00 | 0.00 | 0.00 | 0.00 |  |
| **Total Development Cost** | 250000 |  | Server | 50,000.00 | 0.00 | 0.00 | 0.00 | 0.00 |  |
|  |  |  | **Total Development Costs** | 250,000.00 | 0.00 | 0.00 | 0.00 | 0.00 |  |
| **Operational Costs** |  |  |  |  |  |  |  |  |  |
| Hardware | 500 |  | Hardware | 500.00 | 500.00 | 500.00 | 500.00 | 500.00 |  |
| Operational labor | 40000 |  | Operational labor | 40,000.00 | 40,000.00 | 40,000.00 | 40,000.00 | 40,000.00 |  |
| **Total Operational Costs** | 40500 |  | **Total Operational Costs** | 40,500.00 | 40,500.00 | 40,500.00 | 40,500.00 | 40,500.00 |  |
|  |  |  | **Total Costs** | 290,500.00 | 40,500.00 | 40,500.00 | 40,500.00 | 40,500.00 |  |
|  |  |  | **PV of Costs** | **276,666.67** | **36,734.69** | **34,985.42** | **33,319.45** | **31,732.81** | **413,439.04** |
|  |  |  | **PV of all Costs** | **276,666.67** | **313,401.36** | **348,386.78** | **381,706.23** | **413,439.04** |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  | **Total Project Benefit Costs** | **-180,500.00** | **70,000.00** | **70,500.00** | **71,000.00** | **71,500.00** |  |
|  |  |  | **Yearly NPV** | **-171,904.76** | **63,492.06** | **60,900.55** | **58,411.88** | **56,022.12** | **66,921.85** |
|  |  |  | **Cumulative NPV** | **-171,904.76** | **-108,412.70** | **-47,512.15** | **10,899.73** | **66,921.85** |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  | **Break Even Point** | **3.81** |  |  |  |  |  |
|  |  |  | **Return On Investment** | **16.19%** |  |  |  |  |  |

**Work Plan**

**Work Breakdown Schedule**

1. Business Modeling
   1. Inception
      1. Understand the business situation 0.75 days
      2. Uncover business process problems 0.50 days
   2. Elaboration
      1. Use case development 3 days III.a.2
   3. Construction
   4. Transition
2. Requirements
   1. Inception
      1. Identify appropriate requirements-analysis technique 0.25 days
      2. Identify appropriate requirements-gathering techniques 0.25 days
      3. Identify functional and nonfunctional requirements II.a.1, II.a.2
         1. Perform JAD sessions 4 days
         2. Perform document analysis 6 days II.a.3.A
         3. Conduct interviews II.a.3.A
            1. Interview project manager 0.5 days
            2. Interview business customers 1 day
            3. Interview other stakeholders 1 day
         4. Observe flea market processes 1 day II.a.3.A
         5. Give questionnaires and analyze results 2 day
      4. Analyze current systems 3 days II.a.1, II.a.2
      5. Create requirements definition II.a.3, II.a.4
         1. Determine requirements to track 0.25 days
         2. Compile requirements as they are elicited 4 days II.a.5.A
         3. Review requirements with project manager 2 days II.a.5.B
      6. Create vision document 2 days II.a.3, II.a.4,

IIa.5

* 1. Elaboration
     1. Finalize vision document 1 day II.a.6
     2. Validate use cases with requirements 1 days I.b.1
  2. Construction
  3. Transition

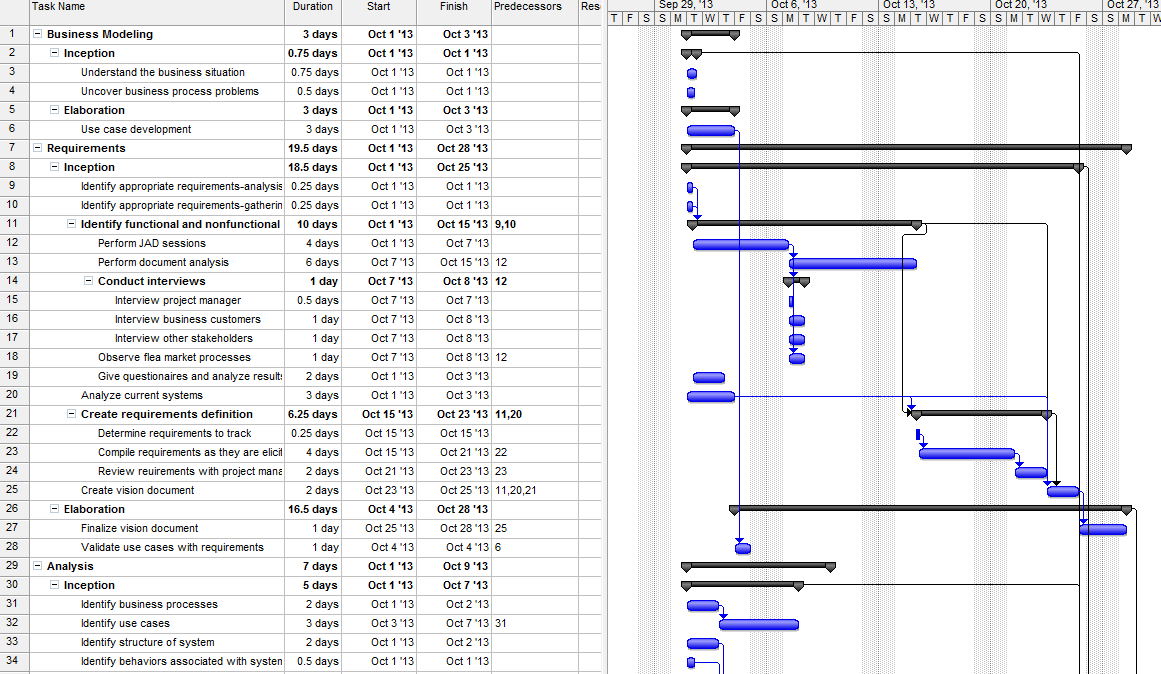
1. Analysis
   1. Inception
      1. Identify business processes 2 days
      2. Identify use cases 3 days III.a.1
      3. Identify structure of system 2 days
      4. Identify behaviors associated with the system 0.5 days
   2. Elaboration
      1. Identify useful reusable classes 1 day
      2. Identify useful class libraries 1 day
      3. Develop UML analysis models 3 day III.a.3, III.a.4
   3. Construction
   4. Transition
2. Design
   1. Inception
   2. Elaboration
   3. Construction
   4. Transition
3. Implementation
   1. Inception
   2. Elaboration
   3. Construction
   4. Transition
4. Test
   1. Inception
   2. Elaboration
   3. Construction
      1. Verify all UML models 3 days III.b.3
   4. Transition
5. Deployment
   1. Inception
   2. Elaboration
   3. Construction
   4. Transition
6. Configuration and Change Management
   1. Inception
      1. Create system to protect and manage project artifacts 4 days
   2. Elaboration
      1. Establish version and control methods 2 days
   3. Construction
   4. Transition
7. Project Management
   1. Inception
      1. Create a workplan for the inception phase 1 day
      2. Create system request 1 day
      3. Perform feasibility analysis
         1. Perform technical feasibility analysis 2 days
         2. Perform economic feasibility analysis 3 days
         3. Perform organizational feasibility analysis 3 days
      4. Identify project effort 1 day
      5. Identify staffing requirements 1 day
      6. Identify cost estimate 1 day
         1. Assess inception phase
      7. Define scope of the system 2 days
   2. Elaboration
      1. Assess elaboration phase 1 day II.b, III.b.VIII.b,
   3. Construction
   4. Transition
8. Environment
   1. Inception
      1. Acquire and install CASE tool 0.25 days
      2. Acquire and install programming environment 0.25 days
      3. Acquire and install configuration and change

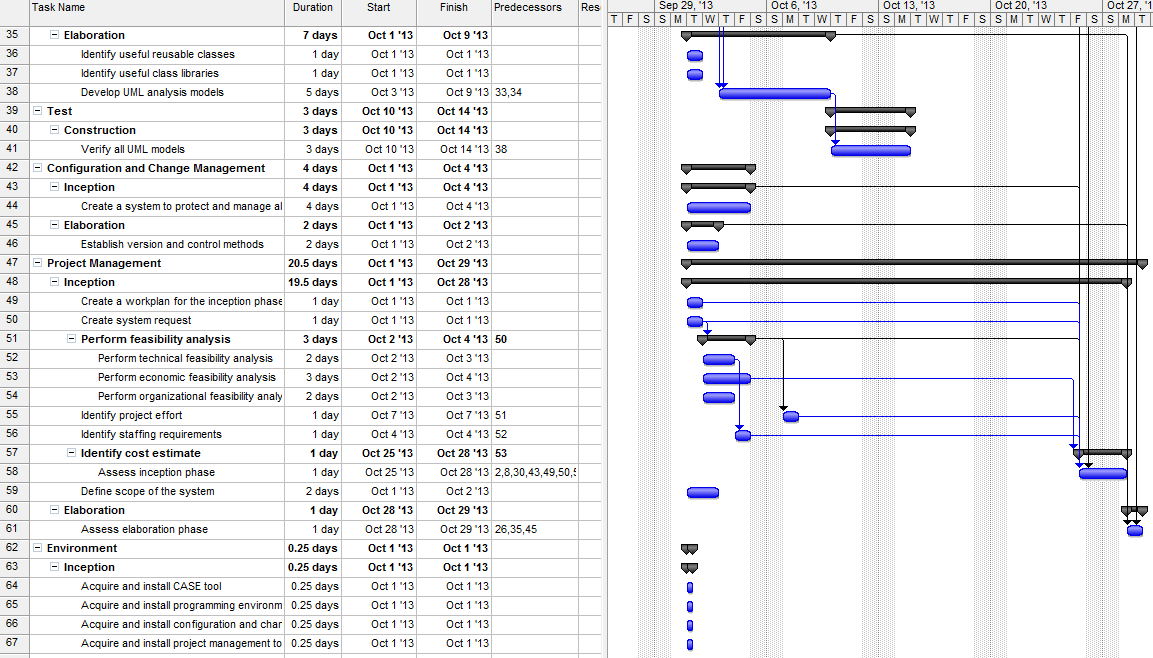
management tools 0.25 days

* + 1. Acquire and install project management tools 0.25 days
  1. Elaboration
  2. Construction
  3. Transition

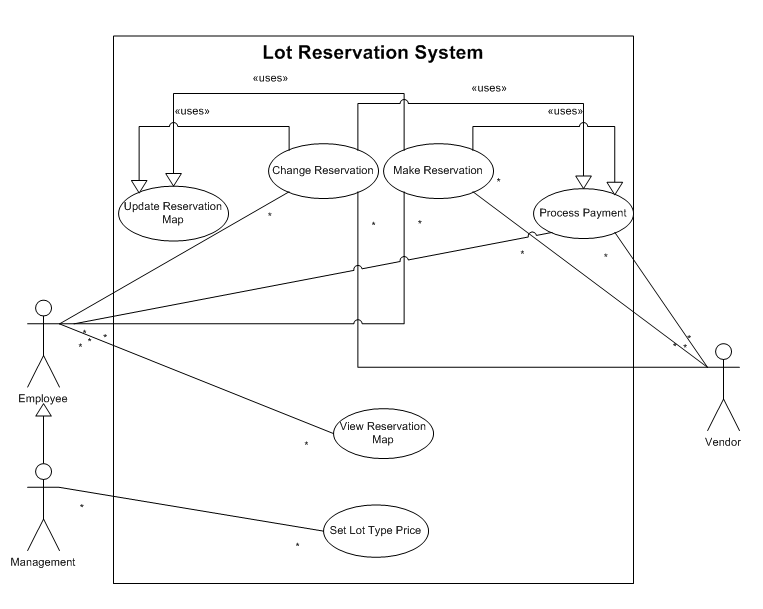
**Use-case Points Analysis**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Unadjusted Actor Weighting Table:** | |  |  |  |  |
| **Actor Type** | **Description** |  | **Weighting Factor** | **Number** | **Result** |
| Simple | External System with a well-defined API | | 1 | 0 | 0 |
| Average | External System using a protocol-based | | 2 | 0 | 0 |
|  | interface, e.g., HTTP, TCT/IP, or a database | |  |  |  |
| Complex | Human |  | 3 | 3 | 9 |
|  |  |  | ***Unadjusted Actor Weight Total(UAW)*** | | **9** |
|  |  |  |  |  |  |
| **Unadjusted Use Case Weighting Table:** | |  |  |  |  |
| **Use-Case Type** | **Description** |  | **Weighting Factor** | **Number** | **Result** |
| Simple | 1-3 transactions |  | 5 | 5 | 25 |
| Complex | 4-7 transactions |  | 10 | 2 | 20 |
| Average | >7 transactions |  | 15 | 0 | 0 |
|  |  | ***Unadjusted*** | ***Use-Case Weight Total (UUCW) =*** | | **45** |
|  |  |  |  |  |  |
| ***Unadjusted Use Case Points (UUCP) = UAW + UUCW =*** | | | **54** |  |  |
|  |  |  |  |  |  |
| **Technical Complexity Factors:** | |  |  |  |  |
| **Factor Number** | **Description** | **Weight** | **Assigned Value (0-5)** | **Weighted Value** | **Notes** |
| T1 | Distributed system | 2 | 0 | 0 |  |
| T2 | Response time or throughput | 1 | 3 | 3 |  |
|  | performance objectives |  |  |  |  |
| T3 | End-user online efficiency | 1 | 4 | 4 |  |
| T4 | Complex internal processing | 1 | 1 | 1 |  |
| T5 | Reusability of code | 1 | 2 | 2 |  |
| T6 | Ease of installation | 0.5 | 3 | 1.5 |  |
| T7 | Ease of use | 0.5 | 3 | 1.5 |  |
| T8 | Portability | 2 | 0 | 0 |  |
| T9 | Ease of change | 1 | 2 | 2 |  |
| T10 | Concurrency | 1 | 0 | 0 |  |
| T11 | Special security objectives included | 1 | 0 | 0 |  |
| T12 | Direct access for third parties | 1 | 0 | 0 |  |
| T13 | Special user training required | 1 | 3 | 3 |  |
|  |  | ***Technical Factor Value (Tfactor)*** | | **18** |  |
|  |  |  |  |  |  |
| ***Technical Complexity Factor (TCF) = 0.6 + (0.01 \* Tfactor) =*** | | | **0.78** |  |  |
|  |  |  |  |  |  |
| **Environmental Factors:** | |  |  |  |  |
| **Factor Number** | **Description** | **Weight** | **Assigned Value (0-5)** | **Weighted Value** | **Notes** |
| E1 | Familiarity with system | 1.5 | 3 | 4.5 |  |
|  | developed process being involved |  |  |  |  |
| E2 | Application Experience | 0.5 | 4 | 2 |  |
| E3 | Object-oriented experience | 1 | 4 | 4 |  |
| E4 | Lead analyst capability | 0.5 | 5 | 2.5 |  |
| E5 | Motivation | 1 | 3 | 3 |  |
| E6 | Requirements | 2 | 3 | 6 |  |
| E7 | Part time staff | -1 | 0 | 0 |  |
| E8 | Difficulty of programming language | -1 | 0 | 0 |  |
|  |  | ***Environmental Factor Value (EFactor)*** | | **22** |  |
|  |  |  |  |  |  |
| ***Environmental Factor (EF) = 1.4 + (-0.03 + EFactor) =*** | | | **0.74** |  |  |
| ***Adjusted Use Case Points (UCP) = UUPC \* TCF \* EF =*** | | | **31.17** |  |  |
| ***Effort in Person Hours = UCP \* PHM =*** | | | **623.46** |  | PHM = 20 |

**Gantt Chart**

****

**Modeling the System**

* **Use cases**
  + **Use case diagram**
* **Use-case descriptions**

|  |  |  |
| --- | --- | --- |
| **Use Case Name:** Change Reservation | **ID:** 1 | **Importance Level:** High |
| **Primary Actor:** Vendor, Employee | **Use Case Type:** Detail, Essential | |
| **Stakeholders and Interests:**  Vendor – wants to change a reservation  Employee – wants to ensure the vendor is able to change a reservation | | |
| **Brief Description:**  This use case describes how we change a reservation in the system. | | |
| **Trigger:** Vendor comes to the business and wants to change a reservation.  **Type:** External | | |
| **Relationships:**  **Association:** Vendor, Employee  **Include:** Update Reservation Map, Process Payment  **Extend:**  **Generalization:** | | |
| **Normal Flow of Events:**   1. The Vendor provides the Employee with his or her information and changes 2. The Vendor’s lot information is updated 3. Process Payment(6) use case executed 4. Update Reservation Map(5) use case executed 5. The Employee returns the results to the Vendor | | |
| **SubFlows:** | | |
| **Alternate/Exceptional Flows:** | | |

|  |  |  |
| --- | --- | --- |
| **Use Case Name:** Make Reservation | **ID:** 2 | **Importance Level:** High |
| **Primary Actor:** Vendor, Employee | **Use Case Type:** Detail, Essential | |
| **Stakeholders and Interests:**  Vendor – wants to make a reservation.  Employee – wants to ensure the vendor is able to make a reservation. | | |
| **Brief Description:**  This use case describes how we make a reservation in the system. | | |
| **Trigger:** Vendor comes to the business and wants to make a reservation.  **Type:** External | | |
| **Relationships:**  **Association:** Vendor, Employee  **Include:** Update Reservation Map, Process Payment  **Extend:**  **Generalization:** | | |
| **Normal Flow of Events:**   1. The Vendor provides the Employee with his or her information and preferences 2. The Employee makes a reservation for the Vendor 3. Process Payment(6) use case is executed 4. Update Reservation Map(5) use case executed 5. The Employee returns results to the Vendor | | |
| **SubFlows:** | | |
| **Alternate/Exceptional Flows:** | | |

|  |  |  |
| --- | --- | --- |
| **Use Case Name:** View Reservation Map | **ID:** 3 | **Importance Level:** High |
| **Primary Actor:** Employee | **Use Case Type:** Detail, Essential | |
| **Stakeholders and Interests:**  Employee – wants to be able to see a visual representation of the reservation map | | |
| **Brief Description:**  This use case describes how the employee views the reservation map | | |
| **Trigger:** Employee decides to view the reservation map  **Type:** External | | |
| **Relationships:**  **Association:** Employee  **Include:**  **Extend:**  **Generalization:** | | |
| **Normal Flow of Events:**   1. The system displays the map | | |
| **SubFlows:** | | |
| **Alternate/Exceptional Flows:** | | |

|  |  |  |
| --- | --- | --- |
| **Use Case Name:** Set Lot Type Price | **ID:** 4 | **Importance Level:** High |
| **Primary Actor:** Management | **Use Case Type:** Detail, Essential | |
| **Stakeholders and Interests:**  Management – wants to set the lot type price based on factors of value, business need, and risk  Vendor – doesn’t want to pay an unreasonable amount of money | | |
| **Brief Description:**  This use case describes how the management sets the lot type price based on his or her own conclusions | | |
| **Trigger:** Management decides to set the lot type price based on some conclusions  **Type:** External | | |
| **Relationships:**  **Association:** Management  **Include:**  **Extend:**  **Generalization:** | | |
| **Normal Flow of Events:**   1. Management logs into the system with a password 2. If the password is incorrect   Execute S-1:Password Incorrect Subflow   1. Management selects the lot type 2. Management sets the price | | |
| **SubFlows:**  S-1: Password Incorrect   1. If Management wants to try again   Return to Normal Flow 2   1. If Management does not want to try again   Exit | | |
| **Alternate/Exceptional Flows:** | | |

|  |  |  |
| --- | --- | --- |
| **Use Case Name:** Update Reservation Map | **ID:** 5 | **Importance Level:** High |
| **Primary Actor:** System | **Use Case Type:** Detail, Essential | |
| **Stakeholders and Interests:**  Employee – wants to have an accurate representation of the reservation map  Vendor – wants to be included on the reservation map | | |
| **Brief Description:**  This use case describes how the reservation map is updated | | |
| **Trigger:** A new reservation(2) has been made, or a change has occurred in an existing reservation(1)  **Type:** Temporal | | |
| **Relationships:**  **Association:**  **Include:**  **Extend:**  **Generalization:** | | |
| **Normal Flow of Events:**   1. The system gets the lot information about the changed or new reservation 2. The system determines what changes have been made 3. The system updates the map according to the changes made | | |
| **SubFlows:** | | |
| **Alternate/Exceptional Flows:** | | |

|  |  |  |
| --- | --- | --- |
| **Use Case Name:** Process Payment | **ID:** 6 | **Importance Level:** High |
| **Primary Actor:** Employee, Vendor | **Use Case Type:** Detail, Essential | |
| **Stakeholders and Interests:**  Employee – wants the vendor to be able to pay  Vendor – wants the payment process correctly | | |
| **Brief Description:**  This use case describes how the payment is processed | | |
| **Trigger:** A vendor has made a new reservation, or made a change to an existing reservation.  **Type:** External | | |
| **Relationships:**  **Association:** Employee, Vendor  **Include:**  **Extend:**  **Generalization:** | | |
| **Normal Flow of Events:**   1. Employee calculates payment 2. If the vendor owes money   Employee collects the money   1. If the employee owes money to the vendor   Give vendor amount due   1. Return results to vendor | | |
| **SubFlows:** | | |
| **Alternate/Exceptional Flows:**  2a: Vendor credit is refused  Offer to make other arrangements  2a1: Vendor refuses to make other arrangements  Exit  2a2: Vendor agrees to make other arrangements  Go to Normal Flow 2 | | |

* **CRC Cards**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Front:** | | | |  |
|  | **Class Name:** Reservation | **ID:** 01 | | **Type:**  Concrete, Domain |  |
|  | **Description:** A part of the system that keeps track of the date and lot number. The reservation updates the information that it contains to the employee. | | **Associated Use Cases:** 1,2,3,5 | |  |
|  | **Responsibilities:**  Update map | | **Collaborators:**  Vendor  Lot  Employee  Map | |  |
|  | **Back:** | | | |  |
|  | **Attributes:**  date | | | |  |
|  | **Relationships:**  **Generalization:**  **Aggregation:**  **Other Associations:** Employee, Lot, Vendor, Map | | | |  |
|  |  | | | |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Front:** | | | |  |
|  | **Class Name:** Lot | **ID:** 02 | | **Type:**  Concrete, Domain |  |
|  | **Description:** A part of the system that contains the type of lot. | | **Associated Use Cases:** 1,2,3,4 | |  |
|  | **Responsibilities:** | | **Collaborators:**  Manager  Reservation  Lot Type | |  |
|  | **Back:** | | | |  |
|  | **Attributes:**  lot Type | | | |  |
|  | **Relationships:**  **Generalization:**  **Aggregation:** Lot Type  **Other Associations:** Manager, Reservation, Lot Type | | | |  |
|  |  | | | |  |

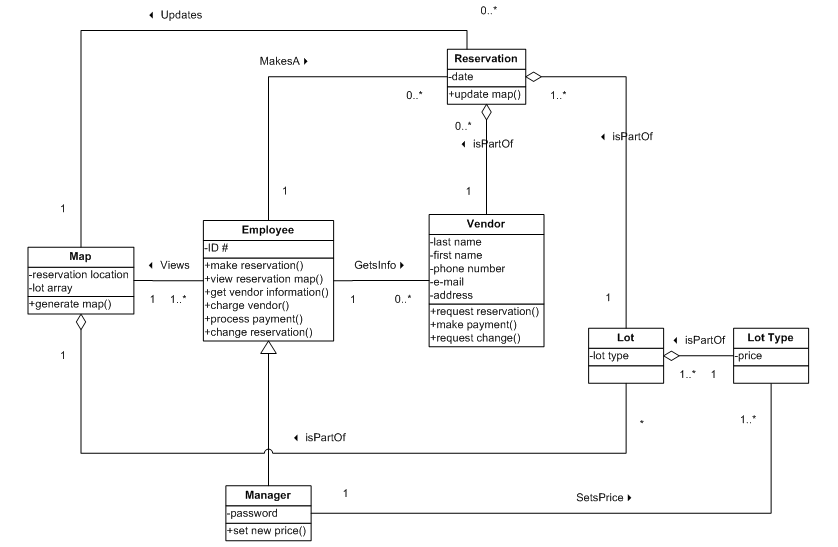
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Front:** | | | |  |
|  | **Class Name:** Vendor | **ID:** 03 | | **Type:**  Concrete, Domain |  |
|  | **Description:** An individual who needs to reserve a lot. | | **Associated Use Cases:** 1,2,6 | |  |
|  | **Responsibilities:**  Request reservation  Make payment  Request change | | **Collaborators:**  Reservation  Employee | |  |
|  | **Back:** | | | |  |
|  | **Attributes:**  last name  first name  phone number  e-mail address  address | | | |  |
|  | **Relationships:**  **Generalization:**  **Aggregation:**  **Other Associations:** Employee, Reservation | | | |  |
|  |  | | | |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Front:** | | | |  |
|  | **Class Name:** Employee | **ID:** 04 | | **Type:**  Concrete, Domain |  |
|  | **Description:** An individual that makes reservations for lots. | | **Associated Use Cases:** 1, 2, 3, 4, 5,6 | |  |
|  | **Responsibilities:**  Make reservation  View reservation map  Get vendor information  Charge vendor  Process payment  Change reservation | | **Collaborators:**  Reservation  Vendor  Manager  Map | |  |
|  | **Back:** | | | |  |
|  | **Attributes:**  ID# | | | |  |
|  | **Relationships:**  **Generalization:**  **Aggregation:** Manager  **Other Associations:** Reservation, Vendor | | | |  |
|  |  | | | |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Front:** | | | |  |
|  | **Class Name:** Manager | **ID:** 05 | | **Type:**  Concrete, Domain |  |
|  | **Description:** An individual who manages the employees and sets the prices for the lots. | | **Associated Use Cases:** 1, 2, 3, 4, 5, 6 | |  |
|  | **Responsibilities:**  Set new price | | **Collaborators:**  Lot Type  Employee | |  |
|  | **Back:** | | | |  |
|  | **Attributes:**  password | | | |  |
|  | **Relationships:**  **Generalization:** Employee  **Aggregation:**  **Other Associations:** Lot Type | | | |  |
|  |  | | | |  |

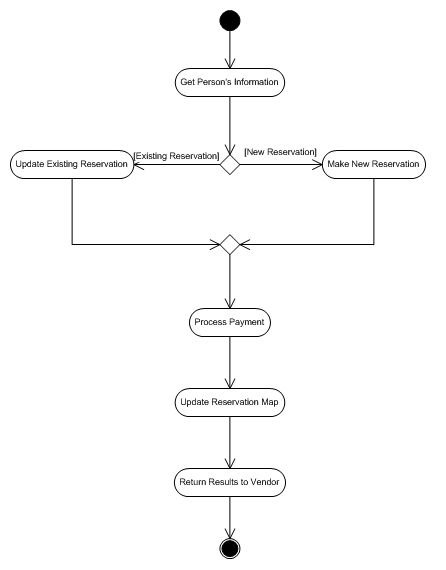
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Front:** | | | |  |
|  | **Class Name:** Lot Type | **ID:** 06 | | **Type:**  Concrete, Domain |  |
|  | **Description:** A part of the system that definitions the price of a lot. | | **Associated Use Cases:** 4, 6 | |  |
|  | **Responsibilities:** | | **Collaborators:**  Lot  Manager | |  |
|  | **Back:** | | | |  |
|  | **Attributes:**  Price | | | |  |
|  | **Relationships:**  **Generalization:** Lot  **Aggregation:**  **Other Associations:** Manager | | | |  |
|  |  | | | |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Front:** | | | |  |
|  | **Class Name:** Map | **ID:** 07 | | **Type:**  Concrete, Domain |  |
|  | **Description:** A part of the system that produces a map of the location of reservations on a lot array. | | **Associated Use Cases:** 2, 3, 5 | |  |
|  | **Responsibilities:**  Generate map | | **Collaborators:**  Reservation  Employee  Lot | |  |
|  | **Back:** | | | |  |
|  | **Attributes:**  Reservation location  Lot array | | | |  |
|  | **Relationships:**  **Generalization:**  **Aggregation:**  **Other Associations:** Reservation, Employee,Lot | | | |  |
|  |  | | | |  |

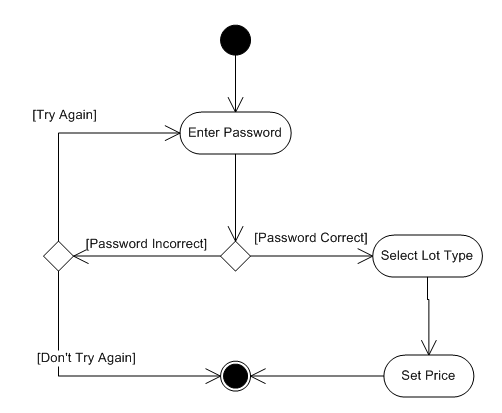
* **Class Diagram**
* **Activity Diagrams**
  + **Justification for Chosen Activity Diagrams**

The activity of updating and making reservations was picked because our system revolves around them. They are the 2 most important functions of our system. The set lot price and process payment activity was picked because it is absolutely essential for the business itself. Without some way for the system to handle monetary needs, the system would be useless to the business. The set lot price and process payment activities are also dependent on one another. The payment could not be processed without the lot price being set, and setting the lot price would be pointless if the payments were not processed. We did not include the update and view reservation map activities because they are very simple. They are also not completely essential for our system or the business.

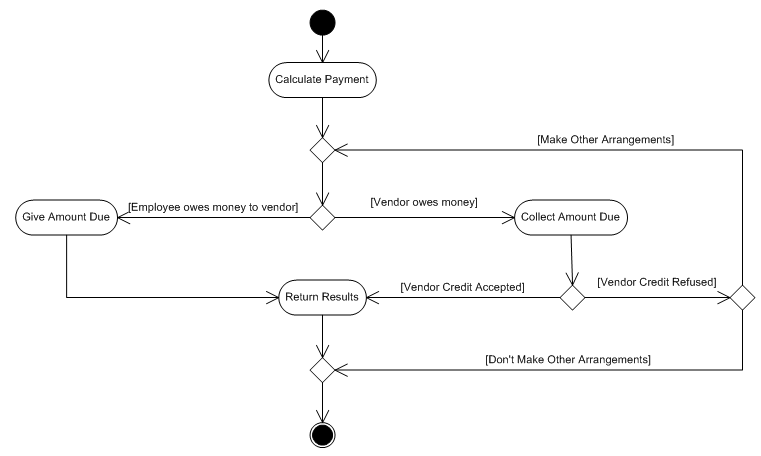
* + **Activity Diagram for Use Case 1: Change Reservation and Use Case 2: Make Reservation**

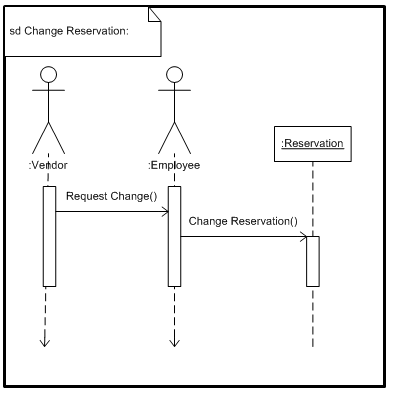
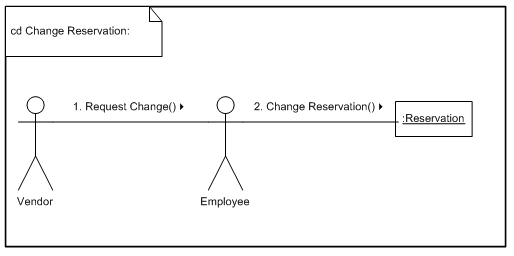
****

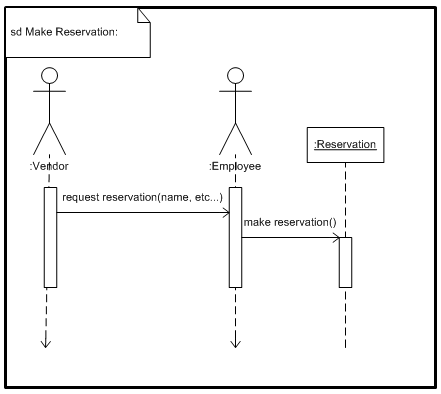
* + **Activity Diagram for Use Case 4: Set Lot Type Price**

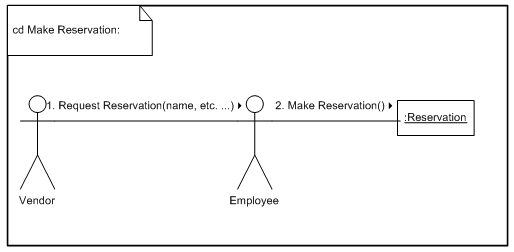
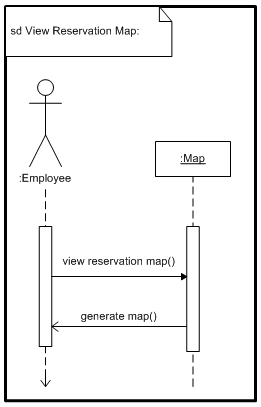
****

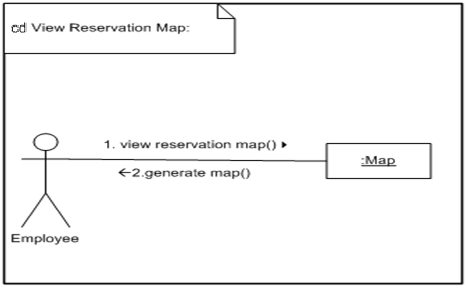
* + **Activity Diagram for Use Case 6: Process Payment**

****

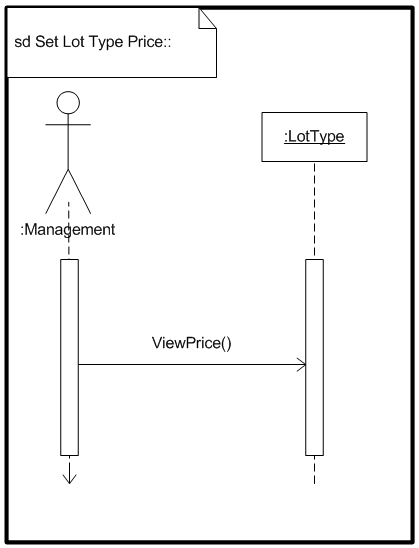
* **Sequence and Communication Diagrams: Generic Diagrams**
  + **Sequence Diagram for Use Case 1: Change Reservation**
  + **Communication Diagram for Use Case 1: Change Reservation**
  + **Sequence Diagram for Use Case 2: Make Reservation**

****

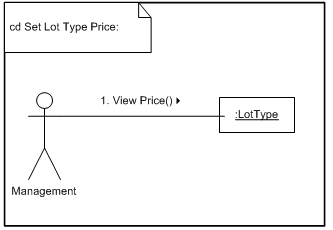
* + **Communication Diagram for Use Case 2: Make Reservation**
  + **Sequence Diagram for Use Case 3: View Reservation Map**
  + **Communication Diagram for Use Case 3: View Reservation Map**

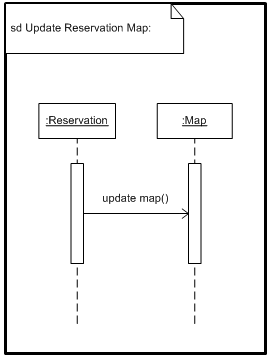
****

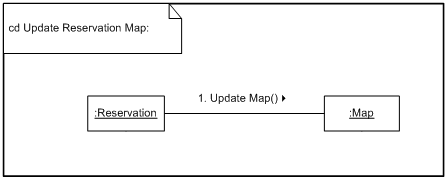
* + **Sequence Diagram for Use Case 4: Set Lot Type Price**

****

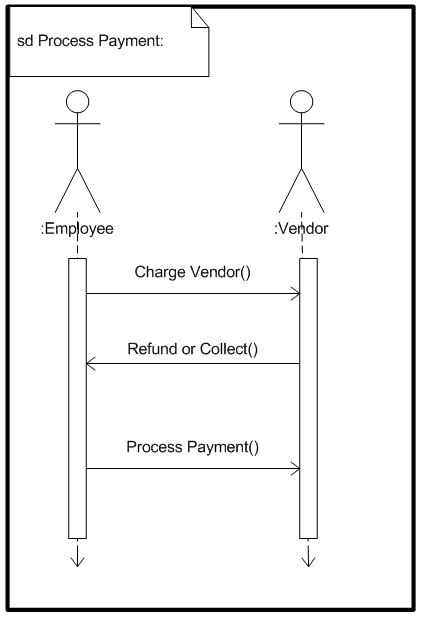
* + **Communication Diagram for Use Case 4: Set Lot Type Price**

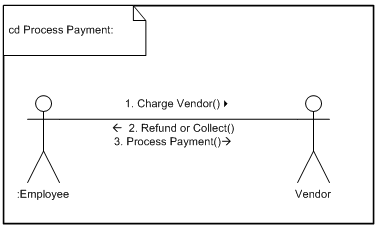
****

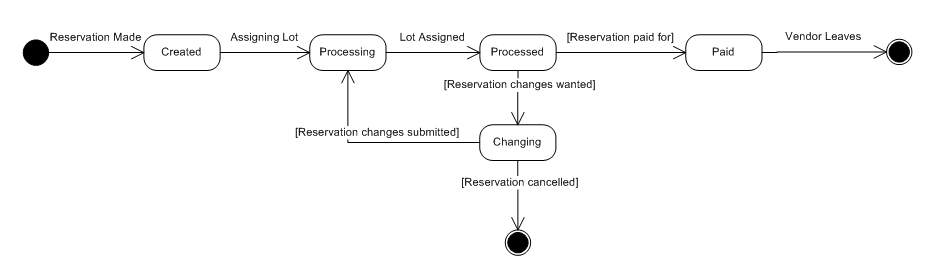
* + **Sequence Diagram for Use Case 5: Update Reservation Map**
  + **Communication Diagram for Use Case 5: Update Reservation Map**

****

* + **Sequence Diagram for Use Case 6: Process Payment**

****

* + **Communication Diagram for Use Case 6: Process Payment**
* **State Machine Diagrams**
  + **Justification for Chosen State Machine Diagram**Since this system is most about the reservation class, we can determine that we must at least have a state machine diagram for it. We only made a state machine diagram for the reservation class object because it is the only complex class in the system. The other classes do not transition through many states in this system.
  + **State Machine Diagram for Reservation Class Object**

****

**User Interface Design**

* **Description of the Interface Standards**
  + Interface Metaphor
    - There is not ONE metaphor for the whole system, but the system contains several metaphors within it.
      * Add Reservation
        + This represents a paper form the employee would fill out and file to add a reservation.
      * Search Results
        + This represents an organized paper list of lots.
      * Reservation Information
        + This represents the same paper as Add Reservation, and would be edited according to the vendor’s wants.
      * Reservation Map
        + This represents a paper map that contains all of the lots and reservations.
  + Interface Objects
    - Buttons are labeled with the action they do.
    - Menus are labeled by content contained.
    - Text Fields contain labels (white text) in them that reflect the content that should be entered into them.
    - Radio buttons have their labels trailing right.
    - Check boxes have their labels trailing right.
  + Interface Actions
    - All actions are labeled using a verb-noun combination or using only a verb.
  + Interface Icons
    - There is only 1. It is a lock on the button Set Lot Price to show that it is password protected.
  + Interface Templates
    - From top to bottom
      * Name of system on border
      * Name of current page
      * Page content display
      * Navigation buttons
    - Color Scheme
      * Black text on white background with blue border
      * Radio buttons and check boxes get a green dot or check mark respectively when selected.
      * Scroll bars have blue arrows and a blue bar on a white background.
      * All objects except labels are outlined in blue.
    - Constants
      * Name of system aligned left on border
      * Name of current page aligned middle
      * Cancel Button exists for every page except the base page (Main Menu) and is aligned right.
* **Windows Navigation Diagram**

<<Window>>  
Main Menu

<<Button>>  
Add Reservation

Add Add Reservation

<<Button>>  
Change Reservation

Add Add Reservation

<<Button>> View Reservation Map

Add Add Reservation

<<Button>>  
Set Lot Type Price

Add Add Reservation

<<Form>>  
Add Reservation

<<Window >>  
View Reservation Map

<<Button>>  
Cancel

Click Add Reservation Button

<<Button>>  
Save

<<Button>>  
Cancel

<<Button>> Delete

<<Form>>  
Reservation Information

Click Save Button

Click Delete Button

<<Button>>  
Cancel

<<Button>>  
Select

Click Cancel Button

Click Change Reservation Button

Click Cancel Button

Click Save Button

<<Button>>  
Save

<<Button>>  
Cancel

Click Cancel Button

Click Cancel Button

Click Select Button

Click Cancel Button

Click Search Button

<<Report>>  
Search Results

Click View Reservation Map Button

<<Form >>  
Change Reservation

<<Button>> Search

<<Button>>  
Cancel

<<Button>>  
Cancel

<<Button>> Select

* **Windows Navigation Diagram (Cont.)**

**\*Copied from above**

Click Cancel/Exit Button

<<Window>>  
Main Menu

<<Button>>  
Add Reservation

Add Add Reservation

<<Button>>  
Change Reservation

Add Add Reservation

<<Button>> View Reservation Map

Add Add Reservation

<<Button>>  
Set Lot Type Price

Add Add Reservation

<<Button>>  
Login

<<Button>>  
Cancel

<<Form >>  
Set Lot Type Price

<<Button>>  
Cancel/Exit

<<Form>>  
Administrative Login

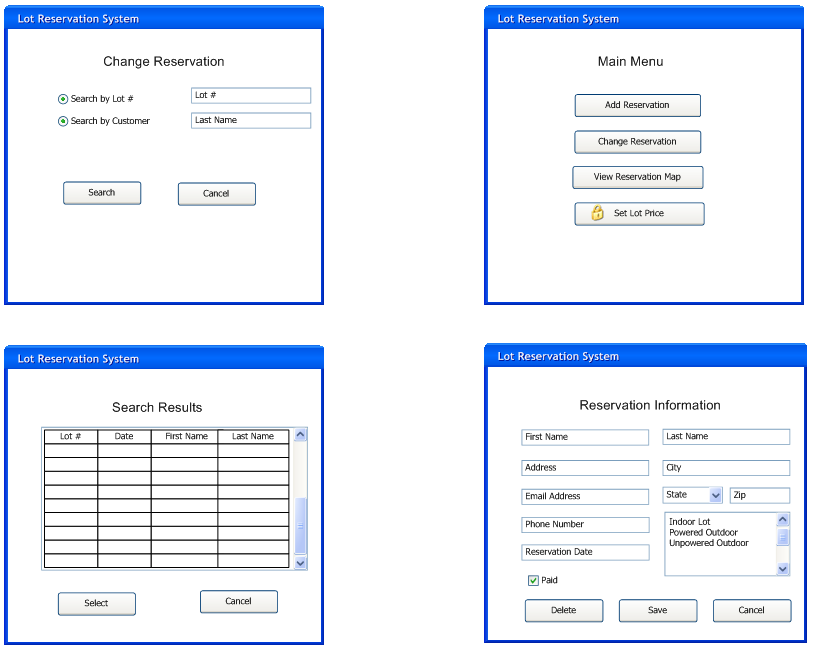
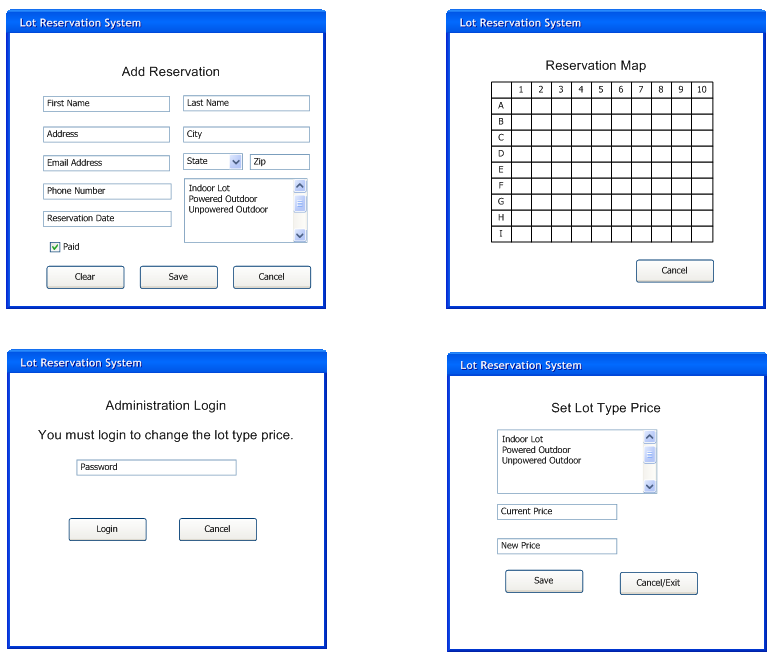
Click Login Button

Click Set Lot Type Price Button

Click Cancel Button

<<Button>>  
Login

<<Button>>  
Cancel

* **Screenshots\Prototypes**
* **Screenshots\Prototypes (Cont.)**
* **Design Choice Justifications**

**Since the users of our system are not very technical, we choose to use buttons in our design to make it easy to navigate the system. We chose the names for the button based on what action the button would perform. Throughout the design of the system, we tried to be consistent in our object’s names and layout. We selected these colors and fonts to make the system visually appealing, easy to read and keep users focus. The text fields in our design include the title of the input desired. This helps the user understand where to input the correct data. We chose to use a drop-drop list on the input for state to reduce incorrect and inconsistent data. A single-select list was chosen for selecting the type of lot because we wanted it easy for the user to remember which types were available. The primary focus for our design was to make the system feel familiar, easy to use, and decrease the amount of input errors.**

**Appendix**

* **Group 4 Meeting Minutes**
  + **Date of meeting 1:** 08/27/2013
    - **Names of those present:** Jeremy Moses, Andrea O’Brien, Chris Womack, and Eric Whatley.
    - **Method of meeting:** Face-to-facein Ayers lounge.
    - **Agenda of meeting:** Times the team are available to meet, team contract, and system ideas.
    - **Old business:** First time meeting. There is not old business.
    - **Discussion summaries:** The team discussed the times we are available to meet. We decided that we are free to meet on Wednesdays at the library on the 10th floor at 2pm each week. We constructed the team contract and all agreed on the terms. We all agreed that Jeremy Moses will be the team leader, and Andrea O’Brien will be the team scribe. We all exchanged contact information and discussed some ideas for the system.
    - **Tasks allocated with deadlines and to whom they are allocated:** There are no deadlines set yet. The only task we have assigned to have the system idea set for next class.
    - **Unresolved issues:** The system idea.
    - **Date and time for next meeting:** After class on 08/29/2013 2pm.
  + **Date of meeting 2:** 08/29/2013
    - **Names of those present:** Jeremy Moses, Andrea O’Brien, Chris Womack, and Eric Whatley.
    - **Method of meeting:** Face-to-face in class.
    - **Agenda of meeting:** Finalize system idea.
    - **Old business:** No old business.
    - **Discussion summaries:** The meeting was short. We all agreed upon the system idea. The system would be a trade-day reservation system. We also discussed the system proposal.
    - **Tasks allocated with deadlines and to whom they are allocated:** Look over other reservation systems that already exist and look over Report I. requirements. We also assigned the system proposal.
    - **Unresolved issues:** Assign sections for Report I. Finish System Proposal.
    - **Date and time for next meeting:** At the library on 09/04/2013 on the 10th floor at 2pm.
  + **Date of meeting 3:** 09/04/2013
    - **Names of those present:** Jeremy Moses, Andrea O’Brien, Chris Womack, and Eric Whatley.
    - **Method of meeting:** Text messaging via cell phone.
    - **Agenda of meeting:** System proposal.
    - **Old business:** No old business.
    - **Discussion summaries:** Since we already had the system proposal completed, we decided that we did not need to meet, and will discuss Report I. in class the next day.
    - **Tasks allocated with deadlines and to whom they are allocated:** None
    - **Unresolved issues:** None
    - **Date and time for next meeting:** 09/05/2013 at 2pm in class.
  + **Date of meeting 4:** 09/05/2013
    - **Names of those present:** Jeremy Moses, Andrea O’Brien, Chris Womack, and Eric Whatley.
    - **Method of meeting:** Face-to-face in class.
    - **Agenda of meeting:** Go over requirements for Report I. Assigning work.
    - **Old business:** Assign sections for Report I.
    - **Discussion summaries:** We met after class and realized that we did not understand the full requirements of the system. We also read through the requirements in report I and talked about who wanted to work on what part.
    - **Tasks allocated with deadlines and to whom they are allocated:** For next meeting the team had to look over other reservation systems that already exist and look over Report I. requirements.
    - **Unresolved issues:** Assign sections for Report I.
    - **Date and time for next meeting:** 09/11/2013 at 2pm on 10th floor of the library.
  + **Date of meeting 5:** 09/11/2013
    - **Names of those present:** Jeremy Moses, Andrea O’Brien, Chris Womack, and Eric Whatley.
    - **Method of meeting:** Face-to-face in the library.
    - **Agenda of meeting:** Go over requirements for Report I. Assigning work.
    - **Old business:** Requirements for Report I. Assigning work.
    - **Discussion summaries:** We print out an outline for the requirements needed in Report I. We discussed what sections would be a fair amount of work per person. We also discussed when each of our sections should be finished for report I. We decided that next Wednesday (the 18th) would be the due date. We decided that Eric would do the description of the business, Andrea would do the scope of the system, and Chris and Jeremy would do the business need, project overview, the stakeholders, and the team configuration.
    - **Tasks allocated with deadlines and to whom they are allocated:** Finish each individual sections of report I. by next meeting (September 18th).
    - **Unresolved issues:** Finish Report I.
    - **Date and time for next meeting:** 09/18/2013 in Library.
  + **Date of meeting 6:** 09/18/2013
    - **Names of those present:** Jeremy Moses, Andrea O’Brien, Chris Womack, and Eric Whatley.
    - **Method of meeting:** Face-to-face in the library.
    - **Agenda of meeting:** Finish Report I. Make individual parts come together.
    - **Old business:** Report I.
    - **Discussion summaries:** Each member discussed their section with each other. We decided if the requirements and contents are necessary. We combined all of the parts into one cohesive report, and made sure that all of the technical requirements are met.
    - **Tasks allocated with deadlines and to whom they are allocated:** Finish report by Friday.
    - **Unresolved issues:** None.
    - **Date and time for next meeting:** 09/20/2013 online. Submit report.
  + **Date of meeting 7:** 09/20/2013
    - **Names of those present:** Jeremy Moses, Andrea O’Brien, Chris Womack, and Eric Whatley.
    - **Method of meeting:** Online/ blackboard.
    - **Agenda of meeting:** Submit report I.
    - **Old business:** None
    - **Discussion summaries:** None
    - **Tasks allocated with deadlines and to whom they are allocated:** None
    - **Unresolved issues:** None
    - **Date and time for next meeting:** 09/25/2013 2pm in library.
  + **Date of meeting 8:** 09/25/2013
    - **Names of those present:** Jeremy Moses, Andrea O’Brien, Chris Womack, and Eric Whatley.
    - **Method of meeting:** Face-to-facein library.
    - **Agenda of meeting:** Report II.
    - **Old business:** None
    - **Discussion summaries:** During this meeting, the group discussed report II. We looked into the feasibility analysis and the WBS. We also assigned work for report II.
    - **Tasks allocated with deadlines and to whom they are allocated:** The report II is divided by: Andrea will do the Technical/ Organizational feasibility and feasibility summary, Chris will do the economic feasibility/summary, Eric will do the WBS outline, Jeremy will do the Gantt chart/ edit report I. Everyone’s work is due by 10/02/13 the next Wednesday, when we meet in the library.
    - **Unresolved issues:** Understanding the WBS.
    - **Date and time for next meeting:** 10/02/2013 at 2pm in library.
  + **Date of meeting 9:** 10/02/2013
    - **Names of those present:** Andrea O’Brien, Chris Womack, and Eric Whatley.
    - **Method of meeting:** Face-to-facein library.
    - **Agenda of meeting:** Finalize sections for report II.
    - **Old business:** Report II.
    - **Discussion summaries:** Worked together to finalize the economic/technical/organization feasibilities. We answered out unresolved issues for each section, and talked about the estimated time frame for he system.
    - **Tasks allocated with deadlines and to whom they are allocated:** Andrea and Chris need to finalize and update their feasibility analysis’ and summaries by the next day in class.
    - **Unresolved issues:** Feasibility analysis.
    - **Date and time for next meeting:** 10/03/2013 at 12:45pm in class.
  + **Date of meeting 10:** 10/03/2013
    - **Names of those present:** Jeremy Moses, Andrea O’Brien, Chris Womack, and Eric Whatley.
    - **Method of meeting:** Face-to-facein Ayers, during class.
    - **Agenda of meeting:** Finish Report II.
    - **Old business:** Report II.
    - **Discussion summaries:** We looked over, and edited each section. The report was put together and the format was corrected.
    - **Tasks allocated with deadlines and to whom they are allocated:** None.
    - **Unresolved issues:** None.
    - **Date and time for next meeting:** 10/04/2013, online, submit report II.
  + **Date of meeting 11:** 10/09/2013
    - **Names of those present:** Jeremy Moses, Andrea O’Brien, Chris Womack, and Eric Whatley.
    - **Method of meeting:** Face-to-facein Library
    - **Agenda of meeting:** Assign Report III/ edit report II.
    - **Old business:** Report II.
    - **Discussion summaries:** We made the use-case diagram and the class diagram rough drafts. We talked about the use-case-point analysis.
    - **Tasks allocated with deadlines and to whom they are allocated:** Jeremy and Eric will work on the use case and the use case descriptions. Andrea will work on the class diagram and the CRC cards. Chris will work on the Use-point analysis. Everyone will edit the part they worked on from the previous report. Everything is due the next Wednesday.
    - **Unresolved issues:** Report III
    - **Date and time for next meeting:** 10/16/2013 2pm
  + **Date of meeting 12:** 10/11/2013
    - **Names of those present:** Jeremy Moses, Andrea O’Brien, Yi Chen, Chris Womack, and Eric Whatley.
    - **Method of meeting:** e-mail
    - **Agenda of meeting:** new member added.
    - **Old business:** none
    - **Discussion summaries:** new member added.
    - **Tasks allocated with deadlines and to whom they are allocated:** None.
    - **Unresolved issues:** None.
    - **Date and time for next meeting:** 10/16/2013 2pm
  + **Date of meeting 13:** 10/16/2013 in Ayers lab at 2pm
    - **Names of those present:** Jeremy Moses, Andrea O’Brien, Yi Chen , Chris Womack, and Eric Whatley.
    - **Method of meeting:** Face-to-facein Ayers, during class.
    - **Agenda of meeting:** Finish Report III.
    - **Old business:** Report III.
    - **Discussion summaries:** Andrea and Yi Chen worked on the CRC cards together, and Eric and Jeremy worked on the Use-Case descriptions. While Chris worked on the Use-point analysis. We answered each others unanswered questions, and decided to have everything done by Monday.
    - **Tasks allocated with deadlines and to whom they are allocated:** Everyone finish their work by Monday.
    - **Unresolved issues:** Report III.
    - **Date and time for next meeting:** 10/22/2013, lab, submit report III.
  + **Date of meeting 14:** 10/22/2013
    - **Names of those present:** Jeremy Moses, Andrea O’Brien, Yi Chen, Chris Womack, and Eric Whatley.
    - **Method of meeting:** Online
    - **Agenda of meeting:** Submit report III.
    - **Old business:** None
    - **Discussion summaries:** None
    - **Tasks allocated with deadlines and to whom they are allocated:** None.
    - **Unresolved issues:** None.
    - **Date and time for next meeting:** 10/23/2013 6pm in Software Lab.
  + **Date of meeting 15:** 10/23/2013 6pm in Software Lab.
    - **Names of those present:** Jeremy Moses, Andrea O’Brien, Yi Chen, Chris Womack, and Eric Whatley.
    - **Method of meeting:** Face-to-face
    - **Agenda of meeting:** Go over requirements for report 4.
    - **Old business:** report 3.
    - **Discussion summaries:** After the group reviews the requirements for report 4, the group decided that a lot of the work would require a lot of collaboration. Andrea, Chris, and Yi would work on the Sequence and communication diagrams. They decided to wait till report 3 is graded so we can decide how to separate the work. The team would further assign work once we receive the graded report 3.
    - **Tasks allocated with deadlines and to whom they are allocated:** no deadlines assigned yet, waiting on the graded report 3.
    - **Unresolved issues:** assigned work.
    - **Date and time for next meeting:**
  + **Date of meeting 16:** 10/27/2013 6pm in Software Lab.
    - **Names of those present:** Jeremy Moses, Andrea O’Brien, Yi Chen, Chris Womack, and Eric Whatley.
    - **Method of meeting:** Face-to-face
    - **Agenda of meeting:** Go over graded report 3, and discuss report 4.
    - **Old business:** report 3
    - **Discussion summaries:**  Andrea, Chris, and Yi split the use cases to do the sequence diagrams and the communication diagrams. Eric decided he wanted to work on the activity diagrams, and Jeremy wanted to do the state diagrams. The group also worked out what screen shots are needed for the interface design. After reviewing the graded report, we the group decided that it was best for the parts that needed editing would be fixed by the group member who previously worked on the part in report 3.
    - **Tasks allocated with deadlines and to whom they are allocated:** The edits for report 3 are to be done by next meeting. Also the team should have parts done by next week.
    - **Unresolved issues:** report 3
    - **Date and time for next meeting:** 10/30/2013 6pm in Software Lab.
  + **Date of meeting 17:** 11/03/2013 6pm in Software Lab.
    - **Names of those present:** Andrea O’Brien, Chris Womack, and Eric Whatley.
    - **Method of meeting:** face-to-face
    - **Agenda of meeting:** Work together on parts for report 4. Go over report 3 edits.
    - **Old business:** report 3
    - **Discussion summaries:** The group as communicated in class about not meeting on the 10/30, we decided to meet on Sunday (11/30) instead. Andrea had issues with the class diagram and consulted Eric for advice about the changes need for the UML diagrams. Eric worked on his use case descriptions and Chris worked on his UML diagrams.
    - **Tasks allocated with deadlines and to whom they are allocated:** Have all parts ready to review for meeting on Wednesday. (11/6)
    - **Unresolved issues:** still editing report 3.
    - **Date and time for next meeting:** 11/06/2013 6pm in Software Lab.
  + **Date of meeting 18:** 11/06/2013 2pm in Software Lab.
    - **Names of those present:** Jeremy Moses, Andrea O’Brien, and Eric Whatley.
    - **Method of meeting:** face-to-face
    - **Agenda of meeting:** Review assigned parts. Complete WMB.
    - **Old business:** report 3.
    - **Discussion summaries:** The groupmet briefly to check on each others work and answer any unanswered questions. Andrea mentioned that she will go to Plotnick to ensure the correctness of the UML diagrams that she has worked on. Eric and Jeremy have completed the interface design.
    - **Tasks allocated with deadlines and to whom they are allocated:** Have everything completed by 11/10/2013
    - **Unresolved issues:** none
    - **Date and time for next meeting:**
  + **Date of meeting 14:** 11/10/2013 6pm in Software Lab.
    - **Names of those present:** Jeremy Moses, Andrea O’Brien, Chris Womack, and Eric Whatley.
    - **Method of meeting:** face-to-face
    - **Agenda of meeting:** finish report 4
    - **Old business:** UML diagrams
    - **Discussion summaries:** After the meeting with Plotnick, Andrea had to update all of her UML diagrams, make changes to the class diagram, and change the CRC cards.Andrea and chris worked together to ensure the correctness of the UML diagrams. Eric and Jeremy worked together on the interface descriptions and other UML diagrams.
    - **Tasks allocated with deadlines and to whom they are allocated:** Go over report 4 before class on Tuesday. Check for errors.
    - **Unresolved issues:** report 4.
    - **Date and time for next meeting:** 10/12/2013 in class.
  + **Date of meeting 14:** 11/12/2013 in class
    - **Names of those present:** Jeremy Moses, Andrea O’Brien, Chris Womack, and Eric Whatley.
    - **Method of meeting:** face-to-face
    - **Agenda of meeting:** submit report 4
    - **Old business:** report 4
    - **Discussion summaries:** The group discussed the changes they have made to report 4, and decided it was okay to submit.
    - **Tasks allocated with deadlines and to whom they are allocated:** none.
    - **Unresolved issues:** report 5.
    - **Date and time for next meeting:** 11/13/2013 2pm in lab.