Chase

Prepared by

Bresia Prudente

Dibyayan Das

Tamina Chatterjee

Karl Consing

Group 9 – Fall 2014

Table of Contents

I Project Description

1 Project Overview

“Chase” is a strategic multiplayer game of chance where each player is a thief who has to travel throughout the United States of America to steal a $400,000 artifact that is due for exhibition, at a museum in Chicago, in a month. The first player to overcome each level’s challenges, arrive at the final destination, and obtain the artifact in the fastest amount of time is declared the winner.

2 The Purpose of the Project

2a The User Business or Background of the Project Effort

The target audience of this game is for people (aged 20 years old or older) who have some experience in business, money, law, and the geography of the United States of America.

“Chase” creates hypothetical scenarios for the player in which he needs to tactfully advance a level without losing his progress, thus testing the individual’s decision-making skills. By gradually recognizing the player’s behavioral pattern based on those decisions, the game will then increase in difficulty as the player progresses.

Traveling is an important feature of the game such that the player must have a fair knowledge of geography. This determines whether the player gets caught or runs out of time and money – two of the most important factors of survival in the game.

With the increasing trend of youth playing computer games, “Chase” is a breath of fresh air as it does not only rely on reaching the next level or overcoming hurdles, but rather, the player’s ability to strategize in order to survive throughout the game.

2b Goals of the Project

We would like to create a product, which not only delivers a unique gaming experience wherein each time the game is played, the situations change, and give different experiences each time. This provides a challenging virtual platform in which the players will have to constantly travel and be pressured to rely on strategy to survive.

2c Measurement

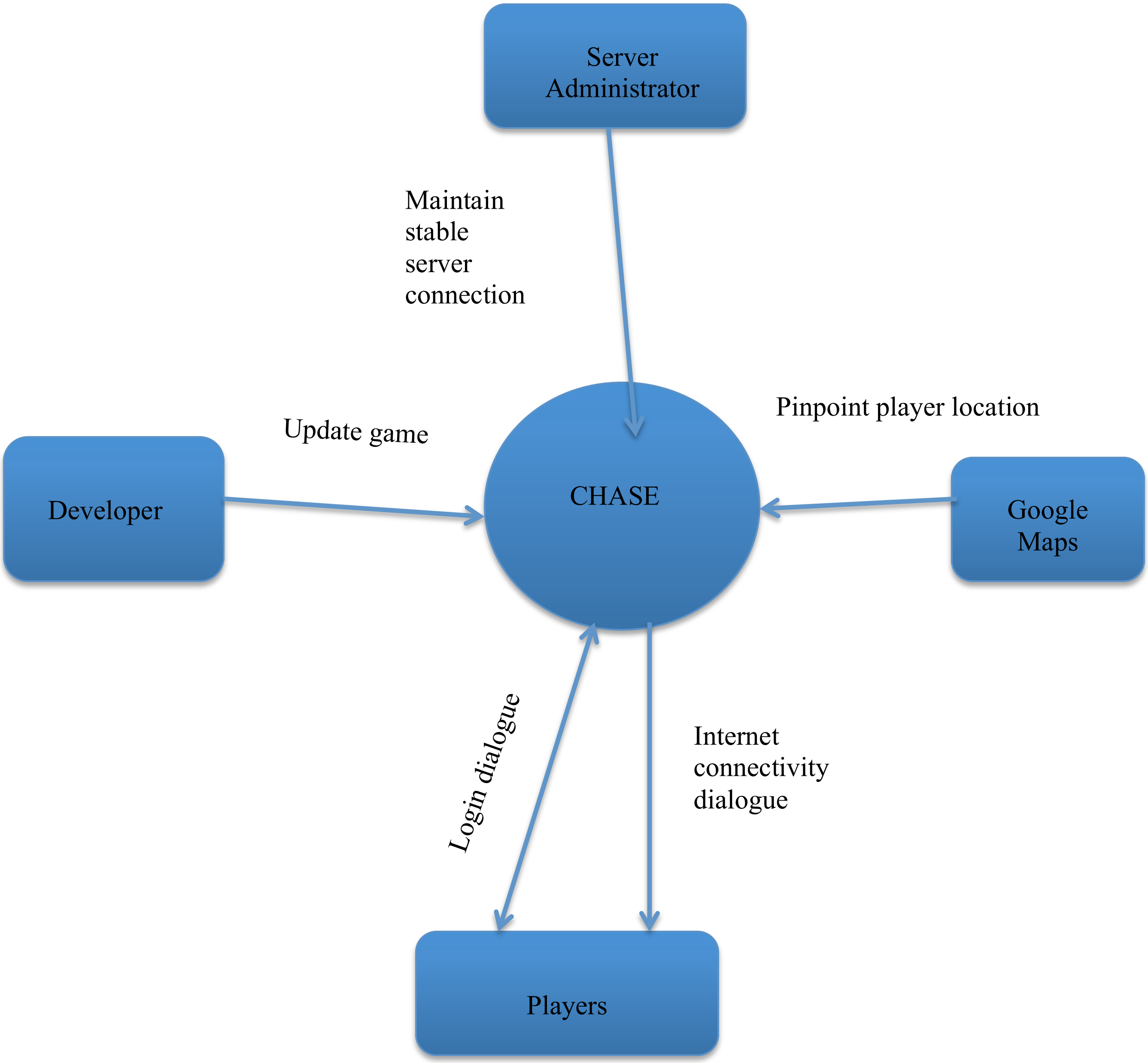
Our project goals can be measured by the number of returning users who were satisfied in their previous experience with the game.

3 The Scope of the Work

“Chase” imitates real life scenarios and environments by incorporating Google Maps for real time positioning of the player on the map and the United States state law for the game's rulebook. The player's progress is saved on the server to be accessed anytime and anywhere.

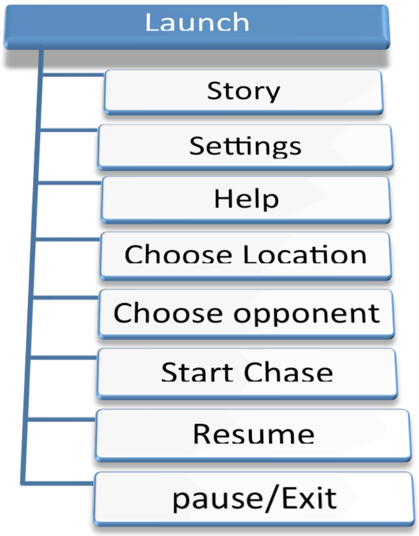
Each player is allowed to choose between existing or new users as opponents, but opponents cannot be changed during the course of the game. Existing players chosen as opponents by the player will be notified and be given the option to decline the invitation. All players can also view details on their opponent's progress and geographical location within the game.

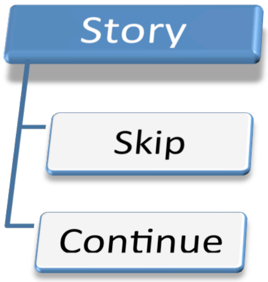
Context Diagram

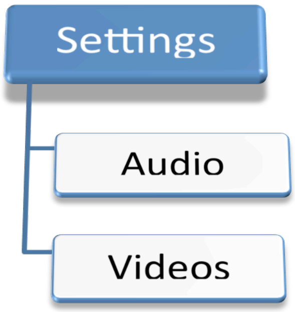


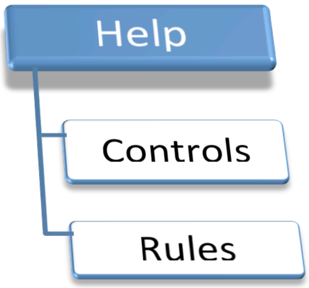
4 The Scope of the Product

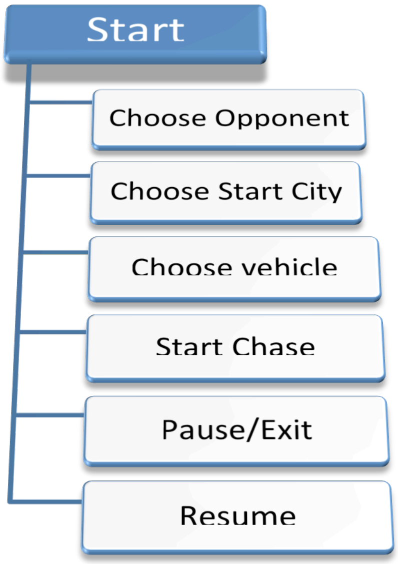
4a Scenario Diagram(s)

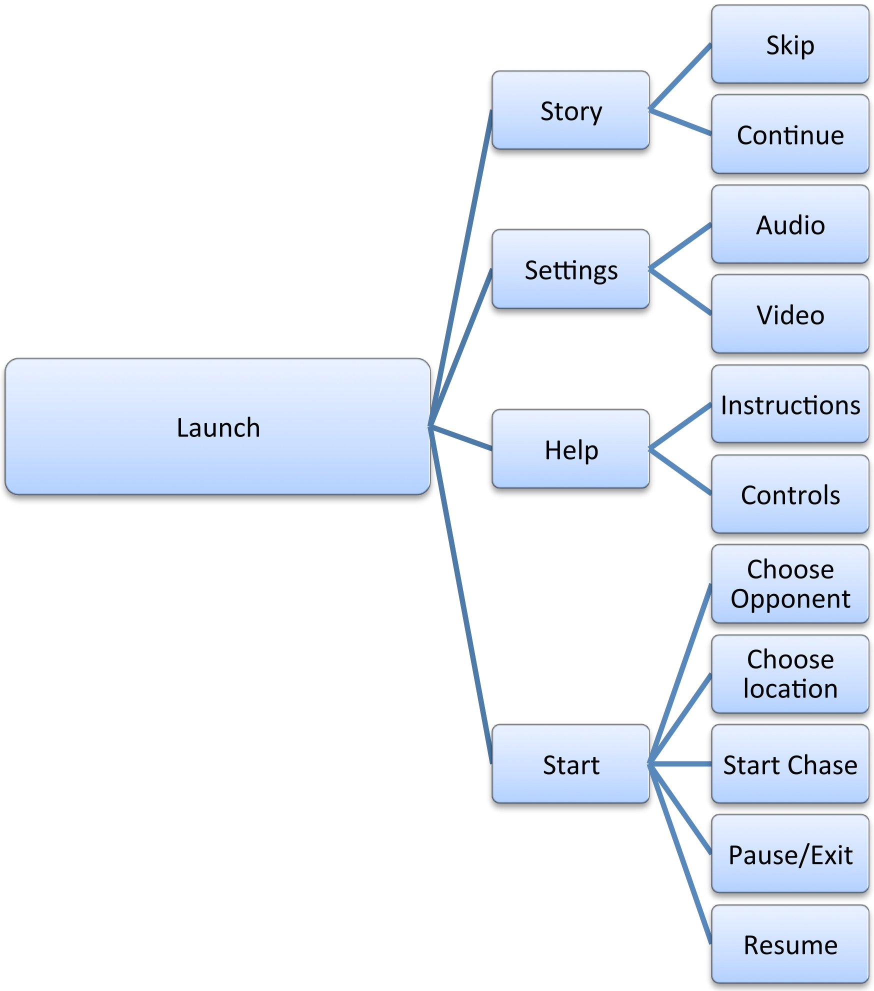









Start
Choose Opponent
Choose Start City
Choose vehicle
Start Chase
Pause/Exit
Resume


Launch
Story
Skip
Continue
Settings
Audio
Video
Help
Instructions
Controls
Start
Choose Opponent
Choose location
Start Chase
Pause/Exit
Resume

4b Product Scenario List

Scenario 1: Launch game

User Action: Launch the game along with a list of servers to join (opponents are selected on a global scale) or locally. If a server is chosen, then a constant net connection is required or else the game can be downloaded for the desktop version. Once the game launches, the player is given a choice to skip the background story of the game or go through it if he chooses to.

Scenario 2: Start Game

User Action: The player starts the game by creating his ID and password, which will become his login credentials for the game.

Scenario 3: Help

User Action: The user can press the help button to know about the rules and controls of the game.

Scenario 4: Select Opponent

User Action: The player is given a choice to select opponents from a group of existing players or have a mock run.

Scenario 4: Select a starting city

User Action: The player needs to go to the select starting city option and select a starting city from the available 0 level cities. This is the 1st location from where the user's journey starts.

Scenario 5: Start chase

User Action: The user needs to press the start button. Once the player starts the game, he is provided an initial amount of money, a fake ID and passport, and is taken to the his chosen starting city.

Scenario 6: Settings

User Action: The user can go to the settings option to configure the audio and display options of the game.

Scenario 7: Resume

User Action: The player can use his login credentials to resume the last known save point.

Scenario6: Exit/Pause

User Action: The player can press the Pause button once to pause the game or press it twice to exit the game.

5 Stakeholders

5a The Client

The client is a game company looking to provide this software to their own customers.

5b The Customer

The customers are those aged 15 or older who frequently plays games.

5c Hands-On Users of the Product

User name/Category: Anyone aged 15 or older who has some knowledge of monetary transactions, law, and geography of the United States of America.

Technological Experience: The system expects the user to have basic computer knowledge like opening a web browser, navigating through the instructions, and selecting appropriate buttons.

Other Specifications: CPU: 2 GHZ Dual Core

RAM: 1 GB

5d Priorities Assigned to Users

The key users of this product will be hardcore gamers. They set aside enough time – ranging from nearly daily to daily- to play this game.

The secondary users will be casual gamers. They play this game frequently or on occasion.

The unimportant users are players who created accounts, but are almost never on or complete disregard the game altogether.

5e User Participation

User participation is somewhat important to the success of the project. Users can provide bug reports as well as any additional features they deem to be necessary to make the game more streamlined.

5f Maintenance Users and Service Technicians

There will be maintenance users who will make sure the servers are constantly running. Additionally, there will be maintenance users who will debug any problems that occur to many users.

5g Other Stakeholders

Other stakeholders are game companies who do not provide the product and players who do not have internet access nor the access to the product.

These are negative stakeholders for the project. They do not need knowledge nor involvement of the project, but are negatively affected by the product.

6 Mandated Constraints

6a Solution Constraints

The product should be a stand-alone executable for the player to play offline. Otherwise, it is an online application in which the user will need to have internet access to in order to seek other players to play competitively. The product seeks other players, either within the country or on a global range from the user's location, and then matches these players for the current user. The user must have a moderately fast internet connection in order to have a streamlined online play.

6b Implementation Environment of the Current System

The environment that the product will be installed is on the player's computer. Regardless of whether the computer is a personal or work computer, the program should work the same. Otherwise, the user needs to have a network connection that is moderately fast and stable.

6c Partner or Collaborative Applications

Google Maps and Google Earth will be one of the collaborative operations.

6d Off-the-Shelf Software

This software is a standalone application if the user chooses to play the software offline.

6e Anticipated Workplace Environment

The user can run the software anywhere as long as the system meets minimum requirements. Once installed into the system, the game can be played without a network connection. However, if the user wishes to play online, there must be a good, stable internet connection in order to access multiplayer.

6f Schedule Constraints

The game development and its testing should meet the planned deadlines.

6g Budget Constraints

The project must be completed within the desired budget of $70,000.

7 Naming Conventions and Definitions

7a Definition of Key Terms

Start Chase – Begin a game of Chase

7b UML and Other Notation Used in this Document

This document generally follows the Version 2.5 OMG UML standard. Any exceptions are noted where used.

7c Data Dictionary for Any Included Models

Currently, there are no dictionary definitions of information flows and stores used in the models.

8 Relevant Facts and Assumptions

8a Facts

“Chase” will be using real time data for geographical positioning of the player. The currency exchange rates, as well as the interest rates, will also dynamically change according to real time market.

The cost of accommodation that the player utilizes will reflect accordingly to the cost of living in the city the player is at that point of time.

8b Assumptions

The game will have a 24-hour access to Google Earth and Google Maps. For an optimized experience, the player must have access to high speed internet. The user must have a system that meets the minimum specifications for the offline version of the game.

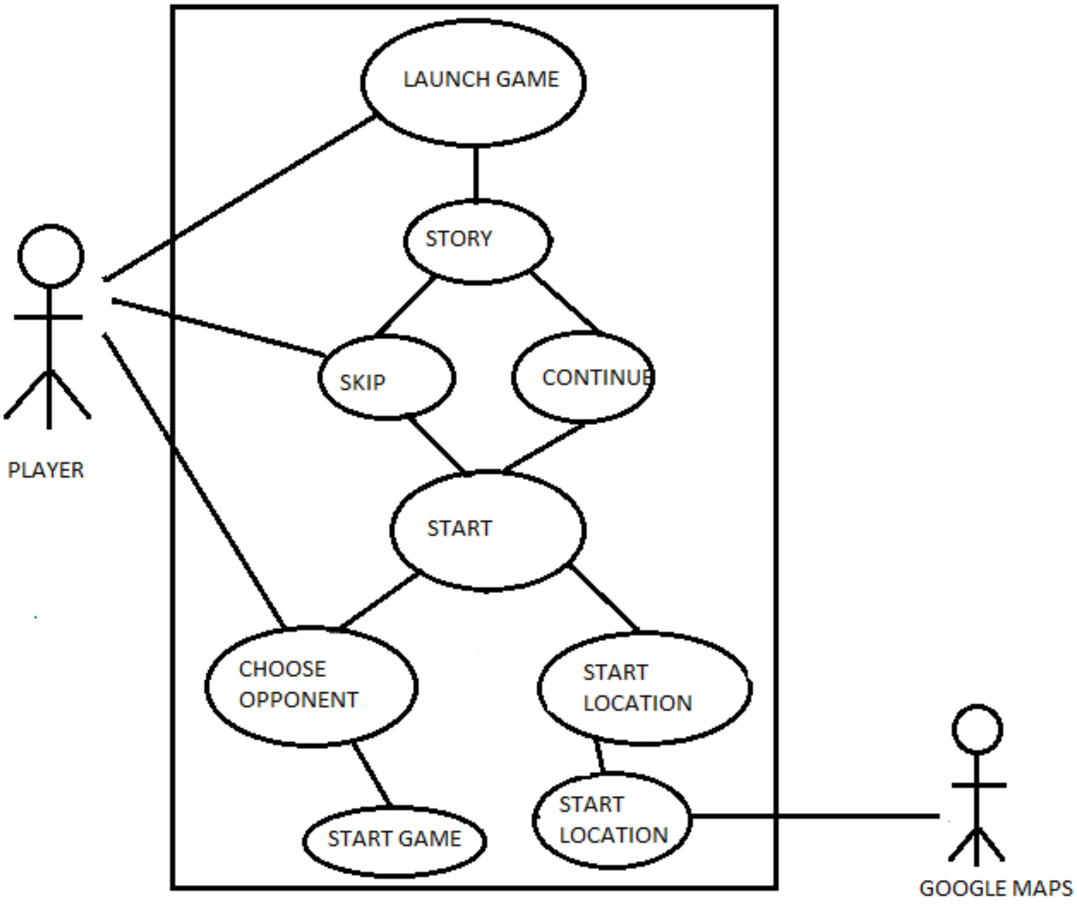
The player has a web browser with the latest versions of Chrome, Firefox or Safari.

The player has a rough idea about trading and directions.

II Requirements

9 Product Use Cases

9a Use Case Diagrams



10 Functional Requirements

1. The game has two versions, an offline and an online version.

   Fit Criterion: The offline version can be downloaded and installed on the user's PC while multiplayer mode is available only in the online version.

2. The game must take the player to the correct location where the user chooses.

   Fit Criterion: The player chooses his starting city from the available level 0 cities. The game begins from the user's chosen location.

3. The game keeps track of each player's monetary information and informs the player the is over when he loses all of his money.

Fit Criterion: The game must update the player's current account balance and when he gains or loses money.

4. The game keeps track of the moves made by the player. In doing so, the game will determine the next situation.

   Fit Criterion: The game provides two options to the player at each hurdle. The decision-making difficulty increases as the game progresses.

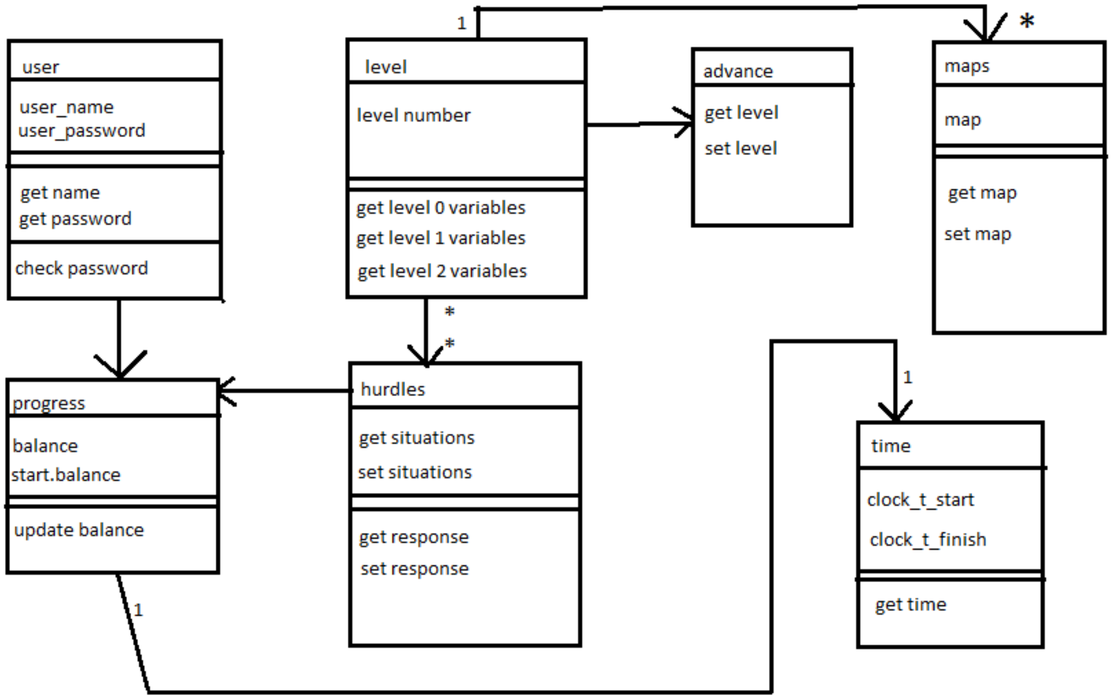
5. The game keeps track of each player's progress so that he can resume from where he left off.

   Fit Criterion: Each time a player logs off or leaves during a game, all progress up to that point will be saved.

6. The game records the time taken by each player to overcome each hurdle and finally reach Chicago.

   Fit Criterion: The game decides the winner based on fastest time and money accumulated by the player.

11 Data Requirements



12 Performance Requirements

12a Speed and Latency Requirements

- During the course of the game, the software must respond to the player's input in no more than one second.
- When the player is thinking or away from the interface, the game must hold the current state.
- The game must run no less than fifty frames per second.
- When loading different screens, game must load screen in less than 2 seconds.
- The animation of the game must run no less than thirty-five frames per second.
- When the player has only less than minimum amount of money required, the game must warn the player to earn more money.
- When artificial intelligence thinks about the next hurdle, it must make its decision in less than 2 seconds.
- Change of cities or levels must happen in less than five seconds.
- If the user chooses to quit, the game must exit immediately.
- The opponent must be chosen for the player in less than 60 seconds for a multiplayer game.

12b Precision or Accuracy Requirements

- The player interacts with the game using his mouse and keypad. The player interface controls have to be fast enough to capture key actions –like the click of a mouse and the press of a key- without frustrating the player with erroneous actions.
- All monetary amounts must be accurate up to two decimal places.
- All cities and their geography should at least be as accurate compared to Google Maps.

12c Capacity Requirements

- The maximum file size of the final game of 50 MB will adhere to the policy of the respective app stores. As for Google Play Store, the Android OS allows a maximum of 50 MB APK application file as where Apple allows a 100 MB for an application file.

- With the limitations of available memory for handheld devices, consequent updates to the game should be kept minimal.

13 Dependability Requirements

13a Reliability Requirements

- For every consecutive 100 runs of the game by the same player, the game should not fail more than twice. (i.e., the failure rate should always be less than 3%)

- In the case of failure, all the player data shall be maintained and restored once the player starts the game again.

- In case of multiplayer game, if one player drops out or is disconnected, his progress in the game and all relevant data is stored and will not affect the other players' game.

13b Availability Requirements

- The game has a dependency on an external product that is a database when played online. Therefore, the online version of the game shall be available 24 hours a day, 365 days a year as long as the player's device is on and connected to the internet and the database is online.

- The game will not be available when updates are being pushed to the game. The game will be available once all updates are successful.

- Updates will take place only when the game begins, not during the player's gameplay.

- If the system crashes, it should be up and running in no more than an hour.

- To avoid crashes, routine maintenance and backups will take place once every month, on the first Sunday of the month, between the hours of 0000 and 0100 PST. The game will not be available in the online mode during this time.

13c Robustness or Fault-Tolerance Requirements

- If the player's device loses power during playing the game, and he connects it immediately, the player should be able to return to the state in which the game was at before the device lost power.

- The game will temporarily switch to the offline mode if connection is lost. It will try to reconnect for 10 minutes before saving his progress for the online mode and permanently switch to the offline mode.

- If the game hangs or crashes, the player's progress is not hampered. He can resume from his last save point.

13d Safety-Critical Requirements

- Since running the game consumes the Central Processing Unit cycles, it often results in the device heating up, which could be a potential cause of damage to the player and the device. Hence, the game should monitor these issues and ensure the device doesn't overheat. It should warn the user of the overheating and pause the game if necessary. It should resume only after it has been connected to an external fan or the device cools down substantially.

- With the limited Random Access Memory available on various devices and multi-tasking capabilities, the game should make sure that it is not consuming greater RAM which may cause 'Out of Memory' errors resulting in abrupt crashes and obstructing the player's ability to multi-task.

- The game, its developers or distributors are not responsible for any harm caused, either physically to the person or to the device due to unsafe environment, improper use or not following standard safety procedures while playing the game.

14 Maintainability and Supportability Requirements

14a Maintenance Requirements

- The game immediately installs updates and patches as they become readily available, so that the game and the player's security are not at risk from various vulnerabilities.

- The game should remind the user of its minimum requirements for proper functioning and usability from time to time.

14b Supportability Requirements

- If a player suggests some bug in the game, it will be a priority and attended to in the next update of the game.

- The game comes with a manual that includes a step-by-step guide on installation and gameplay.

- Any other problems to the game can be reported to the game's 24-hour customer care center over the phone by dialing 1-800-TOCHASE.

- Player feedback will play a crucial role in future upgrades to the game.

- Player queries should be responded to within 48 hours.

- Player suggested enhancements and features should be considered for future updates to the game.

14c Adaptability Requirements

- The game will run on all latest versions of Linux and Windows including Windows XP, Vista, 7, 8 and 8.1.

- The game also has support that will extend to any upcoming versions of Windows and Linux.

- The game will mainly target audiences in North America and Europe and then expand to reach Asian markets.

  Fit Criterion: Minimum System Requirements: 1 GB RAM, at least 1 GB disk space to install the offline version, 256 MB graphics card, functional keyboard, mouse and display, DirectX compatible sound card.

14d Scalability or Extensibility Requirements

- The game server should be able to handle 100,000 online gamers at a time. This number can increase depending on the game's popularity up to 500,000 within 6 months of its launch.

14e Longevity Requirements

- The game will be good for at least 3 years from the date of purchase or the product is eligible for a refund.

- The game will be tested extensively before being released in the market to make sure there are no issues with it.

15 Security Requirements

15a Access Requirements

- Only administrators will have direct access to the servers.

- Only developers will have access to the source code.

- Only valid customers who purchase the product will be provided the product key which will enable them to play the game.

15b Integrity Requirements

- All data critical to the survival of the game is backed up daily and stored in case of accidents.

- The player's personal information is confidential and kept safe.

15c Privacy Requirements

- The game stores the user's personal information securely in a separate database.
- The game informs the user as to what happens to the personal information they provide and how it is used.
- The game shall never reveal any kind of personal information to any outside entity.

15d Audit Requirements

- The game keeps a record of all the people who have used the product at some point of time.

15e Immunity Requirements

- The game is protected by Avast! Antivirus software plugin that prevents malicious attacks on the game.
- The servers are monitored continuously for any malicious activity and a fail-safe is in place for the databases that stores personal user information and critical game data.

16 Usability and Humanity Requirements

16a Ease of Use Requirements

- The game is can be played by anyone over the age of 15 years.

  Fit Criterion: 95% of a test panel of 15 year olds was able to successfully complete CHASE.

- The game requires all users to have a substantial knowledge of money and geography.
- The game will help users avoid making mistakes.

  Fit Criterion: One month's use of the product will result in an error rate of less than 1%.

16b Personalization and Internationalization Requirements

- The game will store the user's custom settings, like his hometown, weather preferences, attire, music and sound levels, brightness levels etc.

- The game is available in these languages: English (US), English (UK), Chinese, Japanese, French, and Spanish. It allows the user to choose a language he is comfortable with.

- The user can personalize the game according to their tastes.

16c Learning Requirements

- The game is easy for any secondary school educated individual.

- The user should have some training in money and geography to properly understand and play the game.

- Any individual with no prior knowledge of the game mechanics can play the game.

16d Understandability and Politeness Requirements

- The game uses places, words and symbols that are naturally understandable by the community.

- The game does not offend any religious or political sentiments.

- The game hides unnecessary details of its build from the user.

16e Accessibility Requirements

- The game is accessible to users with hearing disability.

- The game conforms to Americans with Disability Act.

16f User Documentation Requirements

- The game contains a user manual to help users become familiarized with the game.

- The game comes with an instruction and service manual in case the user experiences difficulty setting up.

- The game also contains an installation guide to help users download and play offline.

16g Training Requirements

- Training is not required to play this game.

- Users should have basic hardware knowledge (such as operating a keyboard or mouse) to begin playing this game.

17 Look and Feel Requirements

17a Appearance Requirements

- **Clarity**: Text should be legible at every size, icons must be precise and lucid, adornments should subtle and appropriate, and a sharpened focus on functionality motivates the design.

- **Depth**: Visual layers and realistic motion impart vitality and heighten players' delight and understanding.

17b Style Requirements

- The game graphics will look and feel realistic.

- The game will give a realistic feeling of travelling and stealing artifacts.

18 Operational and Environmental Requirements

18a Expected Physical Environment

- The game should be played in a room with access to a power source along with sufficient cooling available for the CPU.

- The game or the device it is being played on should not have contact with any kind of food or beverages.

18b Requirements for Interfacing with Adjacent Systems

- The game will run on the last eight releases of all the latest browsers like Safari, Chrome, Firefox, IE and Opera.

- The game will be supported even if the user's system receives an update or changes platform as long as that platform is supported by the game.

18c Productization Requirements

- The game will be distributed on a DVD and as a digital copy to purchase and download.

- The game can be easily installed without any support by the DVD because of the autorun feature.

- The game should install in about 10 – 20 minutes depending on the system specifications.

18d Release Requirements

- A new version is expected to be released once every year.

- The new release will not render the previous versions useless. It will still have support for all previous versions.

19 Cultural and Political Requirements

19a Cultural Requirements

- The game will not offend any religious or ethnic groups.
- The game will be sensitive to the various diverse markets it is being launched in.
- The game will be able to handle the local time across all time zones in the United States of America.

19b Political Requirements

- The game will be developed and manufactured in the United States of America.

20 Legal Requirements

20a Compliance Requirements

- Products implicitly use patent protected designs or concepts.
- The player will be notified of the End User License Agreement.
- The player will be notified that the network will be accessed when updating the game.

20b Standards Requirements

- The game will comply with the standards of Windows and Linux gaming conventions.
- The game will comply with the ESRB requirements for an “E” rating.

III Design

21 System Design

21a Design goals

The design goals for Chase are carefully selected and simple enough so that it is easy for the developers to make clear and concise design decisions. Our main focus for Chase is accuracy, which is represented in all of the design goals. In short, we want a simple, high-performing, and responsive game with good visuals. We want to bring our players a unique experience while including familiar concepts (i.e. Google Maps, Politics). In order to do this, our design must be manageable and understandable.

- **Efficiency:** Chase consumes few resources while maintaining fast performance. That is, it is cost-effective and high performing.

- **Understandability:** In-game controls and instructions of Chase are well defined and simple so that it has an ease of learning, remembering, and using. The game has good documentation and is user friendly.

- **Portability:** May be transported from platform to platform. Game can be played on multiple platforms. The game may be backward-compatible

- **Maintainability / Extensibility:** The game has traceable requirements and well-defined interfaces. We want to make sure that Chase can be modified and updated easily in case of faults. Chase should also have an object-oriented architecture that enables customizations if we want to have updated versions of the game.

- **Security:** Chase provides protection for users' information and follows various security guidelines and protocols to make users feel guarded when playing the game.

- **Availability:** Chase should be available for its players almost anytime. When a user wants to play the game, they can pick it up and play it with ease.

- **Reliability:** Chase should be designed so that it has minimum bugs and glitches during gameplay. Hence, the game will be heavily tested so that we avoid as many negative feedbacks as we can.

22 Current Software Architecture

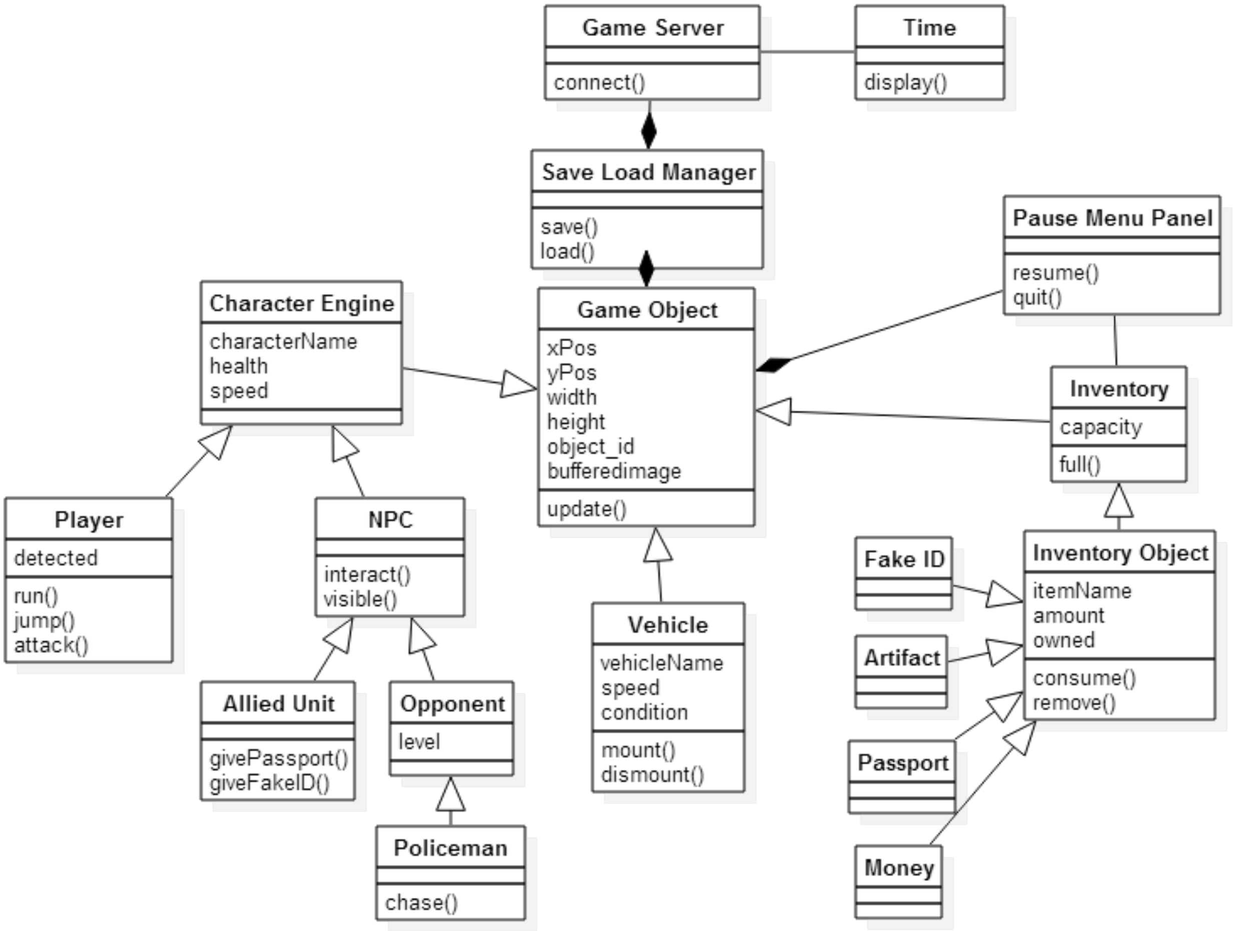
Chase is a new game idea, and we will not be reusing any current software Architecture. The proposed software architecture will be outlined in the next section.

23 Proposed Software Architecture

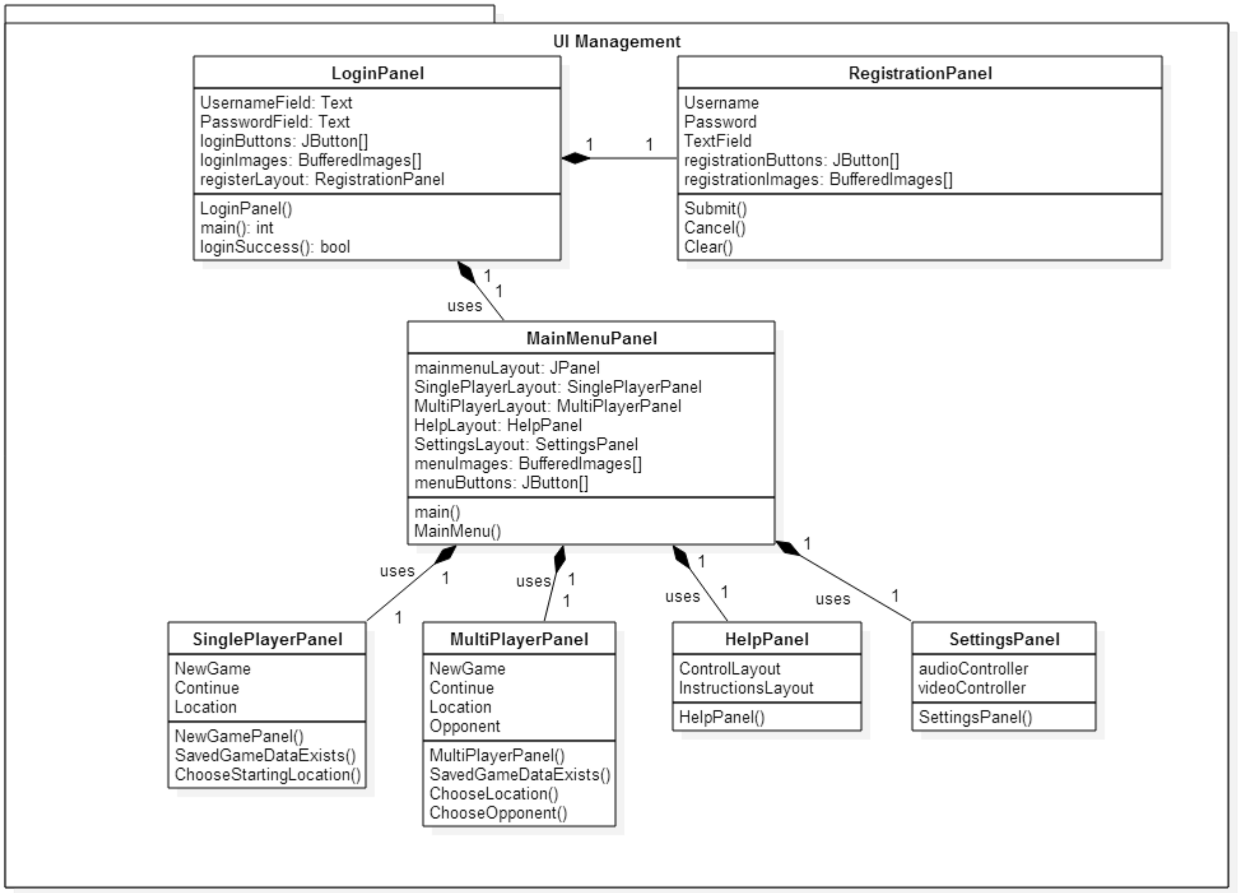
23a Overview

We have decided that Chase will follow a three-tier design architecture because among all design patterns and architectures, the three-tier system fits best. It allows us to organize the subsystems and layers so that our game can easily accommodate change; which, for us, is the most important feature of designing. The architecture will consist of a presentation tier, a logic tier, and finally a data tier. The Presentation tier is accountable for the interface classes between user and the system. The logic layer dives into the 'gameLogic' package. It is responsible for the user's in-game decisions and will show how the game's packages and classes interact. Finally, the data layer is responsible for data and file management. It is composed of 'Data Management Utility' classes, which contain operations for storing various data. For example, saving and loading game states. It is also responsible for knowing where game objects and attributes are located and how they can be called.

23b Class Diagram

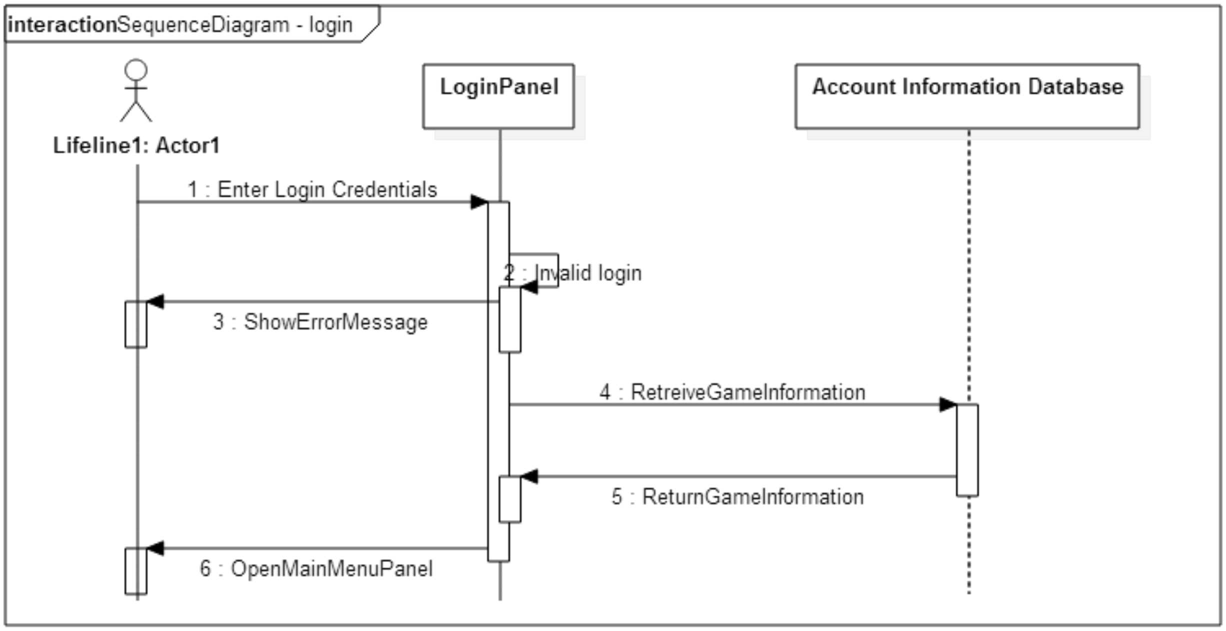


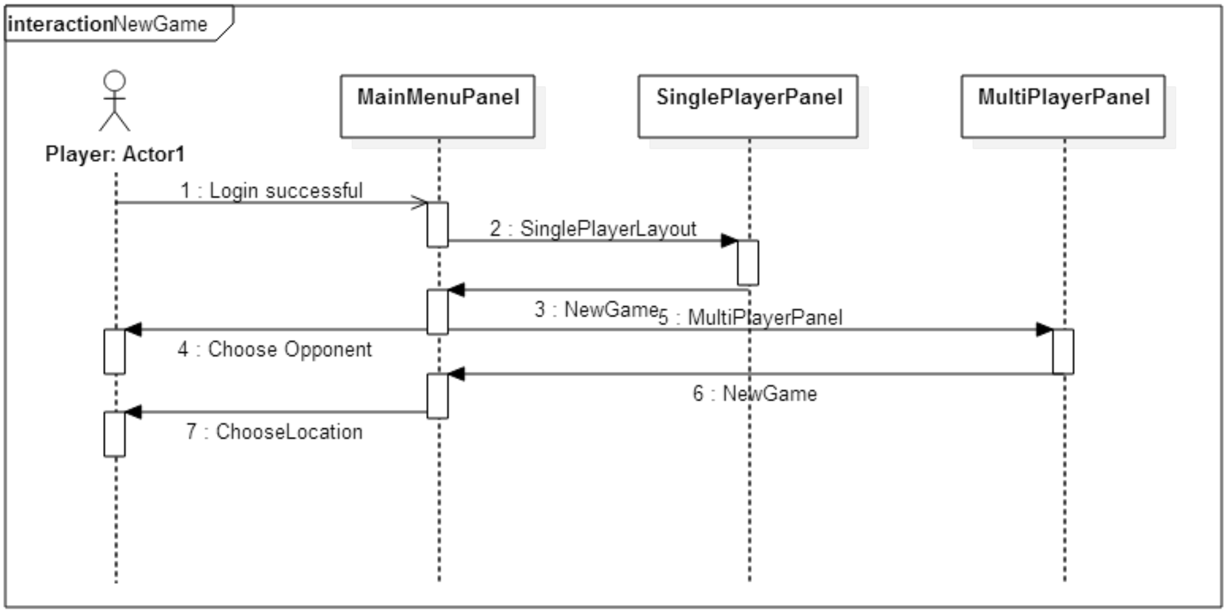
Chase Class Diagram of Game Components



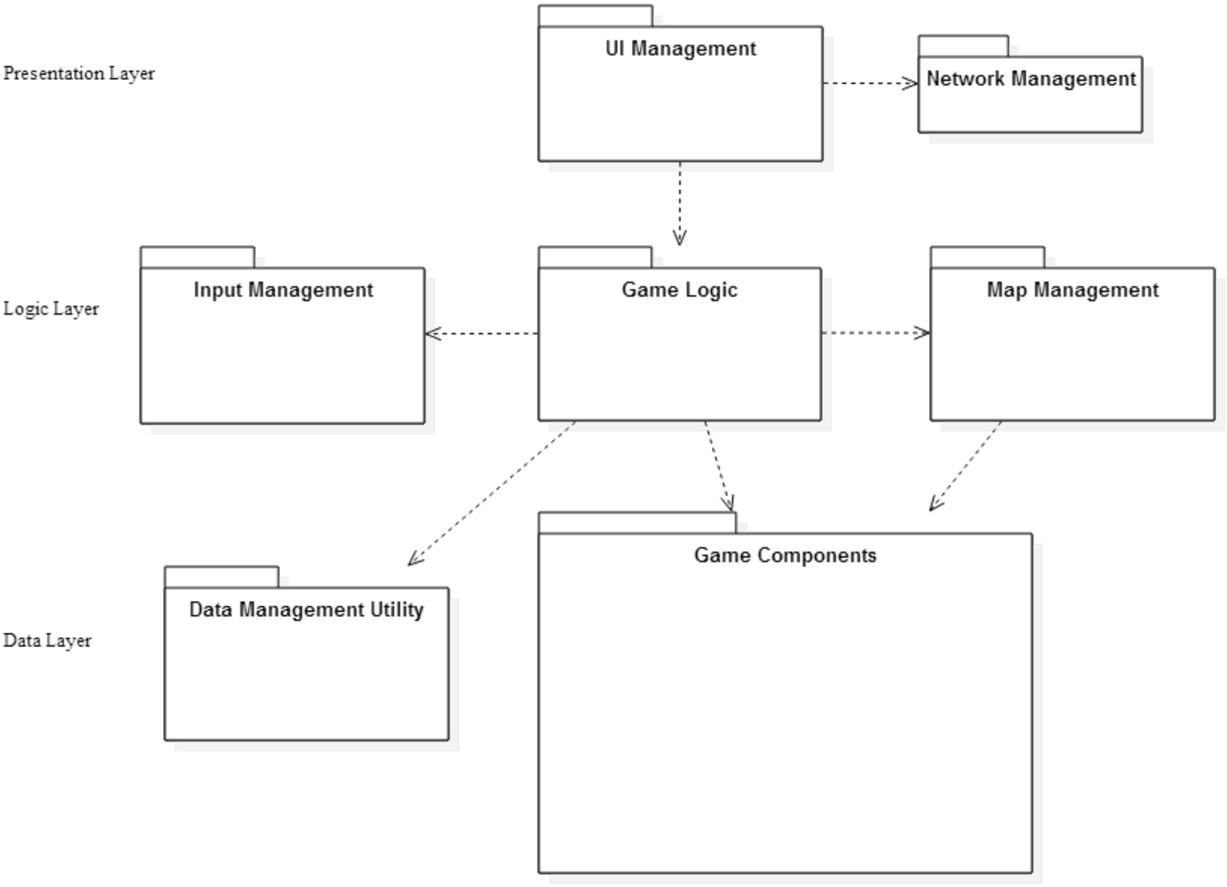
Chase Class Diagram in UI Management Package

23c Dynamic Model


interactionSequenceDiagram - login
Lifeline1: Actor1
LoginPanel
Account Information Database
1 : Enter Login Credentials
2 : Invalid login
3 : ShowErrorMessage
4 : RetreiveGameInformation
5 : ReturnGameInformation
6 : OpenMainMenuPanel



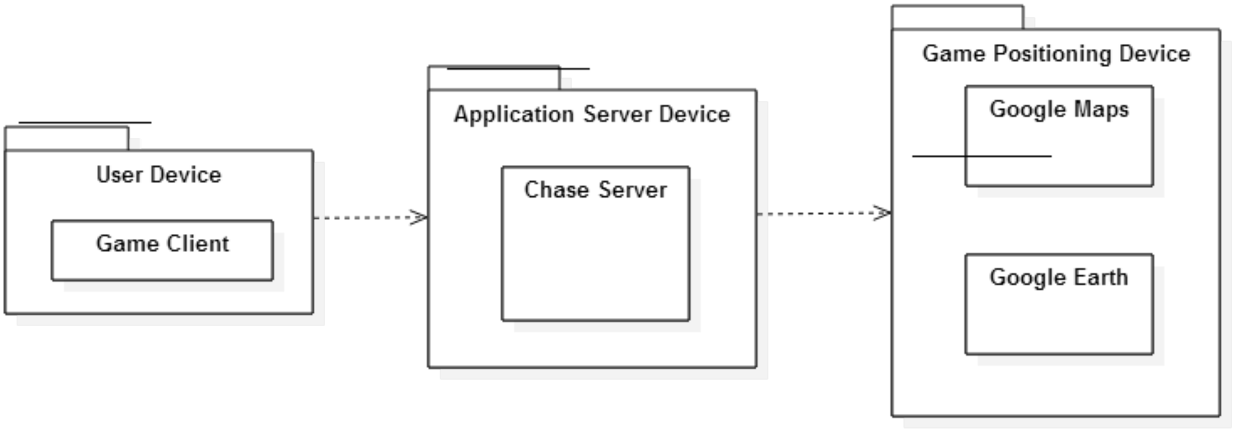
23d Subsystem Decomposition



Chase High-Level Subsystem Decposition

The presentation tier consists of the UI Management Package connected to the Network Interface Package. The logic tier consists of the Input Management Package, and the Map Management Package which both interact with the Game Logic Package. The Game Logic Package contains the façade class, which is the 'gameEngine' class. The data tier consists of the Data Management Package and the Game Components package.

23e Hardware / software mapping



Hardware/Software mapping for Chase

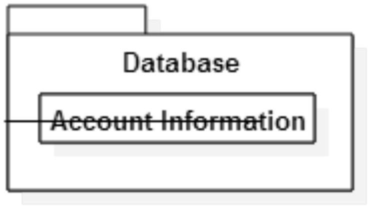
Chase is launched from the user's device. The user will then be prompted to login with their credentials contained in a database in the 'Chase Server'. The game will also grab position information from external sources such as 'Google Maps' and 'Google Earth'.

Hardware requirements are listed in section 5c.

Software requirements are the latest versions of Chrome, Firefox, or Safari.

The game will also run on all latest versions of Linux and Windows including XP, Vista, 7, 8, and 8.1

23f Persistent Data management



Most of the data files and objects used for chase will be downloaded and stored in to the user machine's hard disk drive. The player's position will be gathered from the Game Positioning Device retrieved from Google Maps and Google Earth. Otherwise, data files will be stored in the hard disk drive. A database will be required to store 'Account Information' which contains user credentials.

23g Access control and security

Chase requires login credentials, therefore, the Chase System will store user credentials, and other user information in a database. The user will be able to view only their account information and will not have access to any other accounts in the database. Chase Operators will be able to view and edit account information stored in the database.

23h Boundary conditions

Startup:

- Launch game client from web browser
- Enter login credentials
- Start Chase

Shutdown:

- Select 'exit' from either Main Menu or Pause screen
- Close the browser window

Error Conditions:

- Invalid username/password
- Hardware related crash
- Invalid web browser version
- Connection error

Exception Handling:

- In the event of invalid login, the game will display an error message and redirect user to password recovery page, which also has a link to registration page.
- In the event of a hardware related crash, the game will shut down.
- In the event of invalid web browser version, the game will display an error message and suggest which versions are correct to run the game.
- In the event of a connection error the game will try to reconnect. If a reconnection cannot be established, the game will resort to offline mode.

24 Subsystem services

User Interface Management

The UI Management is what the user will see. It has all the visual and display components such as the MainMenuPanel, SettingsPanel, RegistrationPanel, LoginPanel, HelpPanel, SinglePlayerPanel, and MultiPlayerPanel.

Game Logic

The Game Logic package contains a ‘gameEngine’ class that handles the game mechanics such as collisions. The ‘gameEngine’ class interacts with the ‘graphics’ subsystem in order to render the correct images on to the screen.

Map Management

The Map Management subsystem is in charge of generating and creating the different maps or levels of the game. A ‘voyage’ genre game requires the player to travel from a starting destination to intermediate destinations and finally an end point. This subsystem contains the ‘StageManager’ class.

Input Management

The Input Management subsystem is responsible to designating inputs that trigger events throughout the game. It contains action listeners that determine which keys and commands the player will use.

Data Management

The data management subsystem contains save and load operations, which will be used to store the user’s game data in the ‘File Manager’ class.

Game Components

When the actual game of Chase begins, the ‘Game Components’ class is in charge of organizing the objects that will be used throughout the game. The ‘game object’ class interacts with the ‘Graphics’ class allowing the game to determine which objects will be displayed on the screen at a given time. The Game Components subsystem contains most of the game’s objects and associations.

25 User Interface

The major user interface classes are the associated with the MainMenuPanel class. The game starts off with the login screen which either links to the registration page or the Main Menu page depending on whether the user’s credentials are contained in the Account Information Database. From there, the user can either enter single player mode, multiplayer mode, help menu, or the settings menu. When starting a new game in the single player mode, the user must select only a starting location. In the multiplayer new game, the user can also select an opponent of their choice. If the user decides to continue from where he/she last left off, the game will load all of the users data stored in the File Manager Class in either single or multiplayer modes. The help menu will allow the user to understand the game's instructions and controls. The settings menu allows the user to adjust their desired audio or video settings. There are no current images of the user interfaces at this time.

26 Object Design

26a Object Design trade-offs

Functionality vs. Usability

The more complex the game is, that is adding more game components, we will have to sacrifice usability and understandability of the game. The ease of learning and use will decrease while the game's functionality will increase making a more extensive game. For Chase, we want it to be very extensive and have a lot of different features. This will ensure the game's uniqueness.

Space vs. Speed

As mentioned above, the game is designed to be extensive. This means that the game will require a lot of space for storage. Some speed will be sacrificed to make sure the game has many features the player may interact with. Also, the multiplayer option is going to require that the game will rely on internet connection. This means more speed may potentially be reduced.

26b Interface Documentation guidelines

Identifier Type	Rules for Naming	Examples
Package	The package name must be a noun with mixed case and with each internal words starting with a upper-case alphabet	menuButtons menuImages
Class	The class name must be a noun with mixed case and with each internal words starting with a upper-case alphabet	registrationLayout multiPlayerLayout

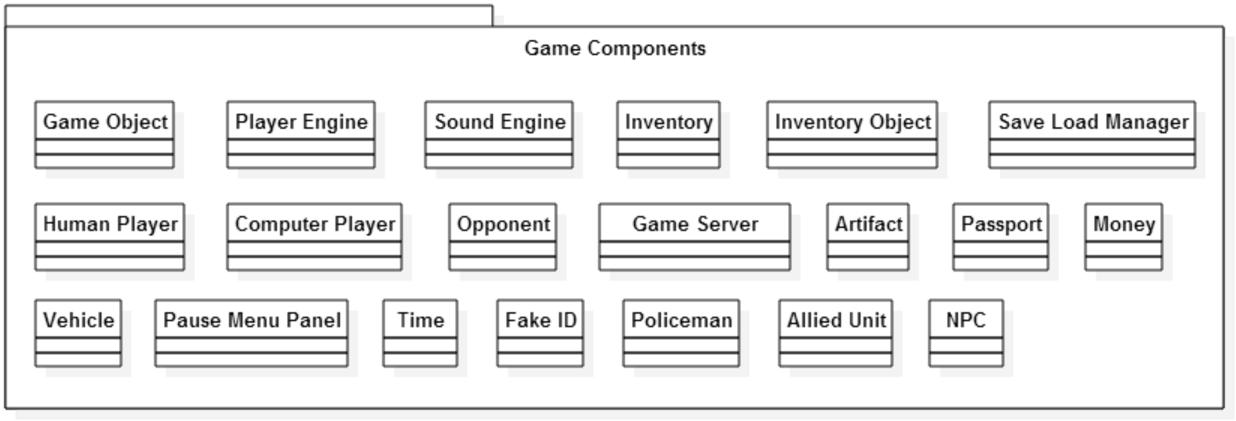
Method	The method name must be a verb with mixed case and with first internal word starting with a lower-case alphabet. Rest of the internal words starting with a upper-case alphabet.	chooseLocation();
		chooseOpponent();

Variable	The variable name must be a noun with mixed case and with each internal words starting with a upper-case alphabet.	string itemName;
		int amount;

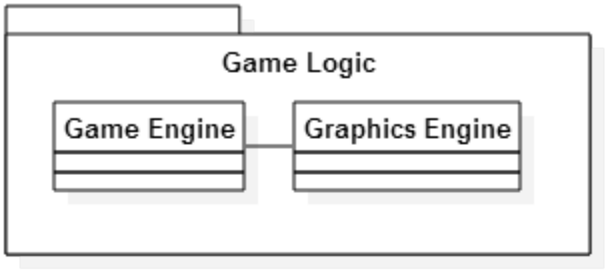
Table: Interface Documentation Guidelines

Table: Interface Documentation Guidelines is reused from the 'Stock Buddy' Design Report from a previous semester. These guidelines are pretty general so we found it useful to reuse their architecture. The examples are changed to fit Chase.

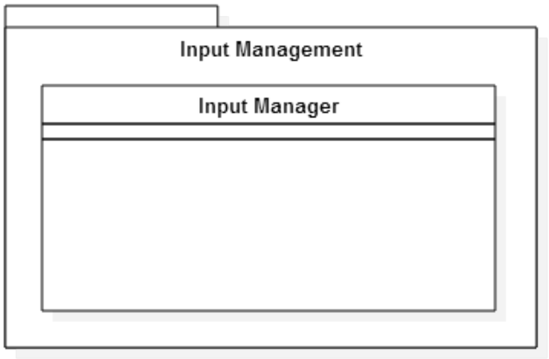
26c Packages



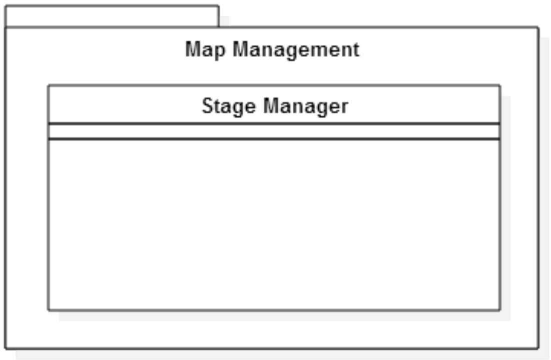
Chase Package Diagram of Game Components



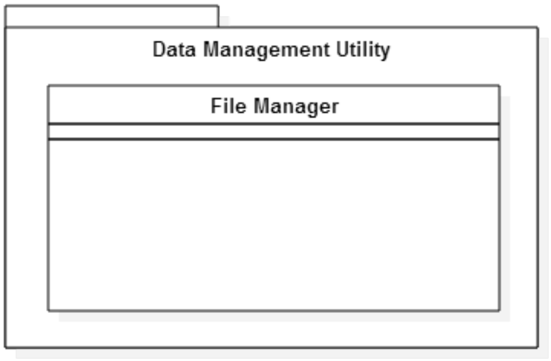
Game Logic Package Diagram



Input Management Package



Map Management Package



Data Management Utility Package

26d Class Interfaces

Game Components Class Interface:

- GameServer
- Time
- SaveLoadManager
- GameObject
- PauseMenuPanel
- Inventory
- InventoryObject
- FakeID, Passport, Money, Artifact
- Player
- NPC
- AlliedUnit
- Opponent
- Policeman

UI Management Class Interface

- LoginPanel
- RegistrationPanel
- MainMenuPanel
- SinglePlayerPanel
- MultiPlayerPanel
- HelpPanel
- SettingsPanel

IV Test Plans

27 Features to be tested / not to be tested

Content

Motivation

Considerations

Example

28 Pass/Fail Criteria

Content

Motivation

Considerations

Example

29 Approach

Content

Motivation

Considerations

Example

30 Suspension and resumption

Content

Motivation

Considerations

Example

31 Testing materials (hardware / software requirements)

Content

Motivation

Considerations

Example

32 Test cases

Content

Motivation

Considerations

Example

33 Testing schedule

Content

Motivation

Considerations

Example

V Project Issues

34 Open Issues

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

Content

A statement of factors that are uncertain and might make significant difference to the product.

Motivation

To bring uncertainty out in the open and provide objective input to risk analysis.

Examples

Our investigation into whether the new version of the processor will be suitable for our application is not yet complete.

The government is planning to change the rules about who is responsible for gritting the motorways, but we do not know what those changes might be.

Considerations

Are there any issues that have come up from the requirements gathering that have not yet been resolved? Have you heard of any changes that might occur in the other organizations or systems on your context diagram? Are there any legislative changes that might affect your system? Are there any rumors about your hardware or software suppliers that might have an impact?

35 Off-the-Shelf Solutions

35a Ready-Made Products

Content

List of existing products that should be investigated as potential solutions. Reference any surveys that have been done on these products.

Motivation

To give consideration to whether a solution can be bought.

Considerations

Could you buy something that already exists or is about to become available? It may not be possible at this stage to make this determination with a lot of confidence, but any likely products should be listed here.

Also consider whether some products must not be used.

35b Reusable Components

Content

Description of the candidate components, either bought from outside or built by your company, that could be used by this project. List libraries that could be a source of components.

Motivation

Reuse rather than reinvention.

35c Products That Can Be Copied

Content

List of other similar products or parts of products that you can legally copy or easily modify.

Motivation

Reuse rather than reinvention.

Examples

Another electricity company has built a customer service system. Its hardware is different from ours, but we could buy its specification and cut our analysis effort by approximately 60 percent.

Considerations

While a ready-made solution may not exist, perhaps something, in its essence, is similar enough that you could copy, and possibly modify, it to better effect than starting from scratch. This approach is potentially dangerous because it relies on the base system being of good quality.

This question should always be answered. The act of answering it will force you to look at other existing solutions to similar problems.

36 New Problems

36a Effects on the Current Environment

Content

A description of how the new product will affect the current implementation environment. This section should also cover things that the new product should *not* do.

Motivation

The intention is to discover early any potential conflicts that might otherwise not be realized until implementation time.

Examples

Any change to the scheduling system will affect the work of the engineers in the divisions and the truck drivers.

Considerations

Is it possible that the new system might damage some existing system? Can people be displaced or otherwise affected by the new system?

These issues require a study of the current environment. A model highlighting the effects of the change is a good way to make this information widely understandable.

36b Effects on the Installed Systems

Content

Specification of the interfaces between new and existing systems.

Motivation

Very rarely is a new development intended to stand completely alone. Usually the new system must coexist with some older system. This question forces you to look carefully at the existing system, examining it for potential conflicts with the new development.

36c Potential User Problems

Content

Details of any adverse reaction that might be suffered by existing users.

Motivation

Sometimes existing users are using a product in such a way that they will suffer ill effects from the new system or feature. Identify any likely adverse user reactions, and determine whether we care about those reactions and what precautions we will take.

36d Limitations in the Anticipated Implementation Environment That May Inhibit the New Product

Content

Statement of any potential problems with the new automated technology or new ways of structuring the organization.

Motivation

The intention is to make early discovery of any potential conflicts that might otherwise not be realized until implementation time.

Examples

The planned new server is not powerful enough to cope with our projected growth pattern.

The size and weight of the new product do not fit into the physical environment.

The power capabilities will not satisfy the new product's projected consumption.

Considerations

This requires a study of the intended implementation environment.

36e Follow-Up Problems

Content

Identification of situations that we might not be able to cope with.

Motivation

To guard against situations where the product might fail.

Considerations

Will we create a demand for our product that we are not able to service? Will the new system cause us to run afoul of laws that do not currently apply? Will the existing hardware cope?

There are potentially hundreds of unwanted effects. It pays to answer this question very carefully.

37 Tasks

37a Project Planning

Content

Details of the life cycle and approach that will be used to deliver the product. A high-level process diagram showing the tasks and the interfaces between them is a good way to communicate this information.

Motivation

To specify the approach that will be taken to deliver the product so that everyone has the same expectations.

Considerations

Depending on the maturity level of your process, the new product will be developed using your standard approach. However, some circumstances are unique to a particular product and will necessitate changes to your life cycle. While these considerations are not product requirements, they are needed if the product is to be successfully developed.

If possible, attach an estimate of the time and resources needed for each task based on the requirements that you have specified. Attach your estimates to the events, use cases, and/or functions that you specified in sections 8 and 9.

Do not forget issues related to data conversion, user training, and cutover. These needs are usually ignored when projects set implementation dates.

37b Planning of the Development Phases

Content

Specification of each phase of development and the components in the operating environment.

Motivation

To identify the phases necessary to implement the operating environment for the new system so that the implementation can be managed.

Fit Criterion

Name of the phase.

Required operational date.

Operating environment components included.

Functional requirements included.

Nonfunctional requirements included.

Considerations

Identify which hardware and other devices are necessary for each phase of the new system. This list may not be known at the time of the requirements process, as these devices may be decided at design time.

38 Migration to the New Product

38a Requirements for Migration to the New Product

Content

A list of the conversion activities. Timetable for implementation.

Motivation

To identify conversion tasks as input to the project planning process.

Considerations

Will you use a phased implementation to install the new system? If so, describe which requirements will be implemented by each of the major phases.

What kind of data conversion is necessary? Must special programs be written to transport data from an existing system to the new one? If so, describe the requirements for these programs here.

What kind of manual backup is needed while the new system is installed?

When are each of the major components to be put in place? When are the phases of the implementation to be released?

Is there a need to run the new product in parallel with the existing product?

Will we need additional or different staff?

Is any special effort needed to decommission the old product?

This section is the timetable for implementation of the new system.

38b Data That Has to Be Modified or Translated for the New System

Content

List of data translation tasks.

Motivation

To discover missing tasks that will affect the size and boundaries of the project.

Fit Criterion

Description of the current technology that holds the data.

Description of the new technology that will hold the data.

Description of the data translation tasks.

Foreseeable problems.

Considerations

Every time you make an addition to your dictionary (see section 5), ask this question: Where is this data currently held, and will the new system affect that implementation?

39 Risks

All projects involve risk—namely, the risk that something will go wrong. Risk is not necessarily a bad thing, as no progress is made without taking some risk. However, there is a difference between unmanaged risk—say, shooting dice at a craps table—and managed risk, where the probabilities are well understood and contingency plans are made. Risk is only a bad thing if the risks are ignored and they become problems. Risk management entails assessing which risks are most likely to apply to the project, deciding a course of action if they become problems, and monitoring projects to give early warnings of risks becoming problems.

This section of your specification should contain a list of the most likely risks and the most serious risks for your project. For each risk, include the probability of that risk becoming a problem. Capers Jones's *Assessment and Control of Software Risks* (Prentice-Hall, Englewood Cliffs, N.J., 1994) gives comprehensive lists of risks and their probabilities; you can use these lists as a starting point. For example, Jones cites the following risks as being the most serious:

- Inaccurate metrics

- Inadequate measurement
- Excessive schedule pressure
- Management malpractice
- Inaccurate cost estimating
- Silver bullet syndrome
- Creeping user requirements
- Low quality
- Low productivity
- Cancelled projects

Use your knowledge of the requirements as input to discover which risks are most relevant to your project.

It is also useful input to project management if you include the impact on the schedule, or the cost, if the risk does become a problem.

40 Costs

For details on how to estimate requirements effort and costs, refer to Appendix C Function Point Counting: A Simplified Introduction

The other cost of requirements is the amount of money or effort that you have to spend building them into a product. Once the requirements specification is complete, you can use one of the estimating methods to assess the cost, expressing the result as a monetary amount or time to build.

There is no best method to use when estimating. Keep in mind, however, that your estimates should be based on some tangible, countable artifact. If you are using this template, then, as a result of doing the work of requirements specification, you are producing many measurable deliverables. For example:

- Number of input and output flows on the work context
- Number of business events
- Number of product use cases
- Number of functional requirements
- Number of nonfunctional requirements
- Number of requirements constraints

- Number of function points

The more detailed the work you do on your requirements, the more accurate your deliverables will be. Your cost estimate is the amount of resources you estimate each type of deliverable will take to produce within your environment. You can create some very early cost estimates based on the work context. At that stage, your knowledge of the work will be general, and you should reflect this vagueness by making the cost estimate a range rather than a single figure.

As you increase your knowledge of the requirements, we suggest you try using function point counting—not because it is an inherently superior method, but because it is so widely accepted. So much is known about function point counting that it is possible to make easy comparisons with other products and other installations' productivity.

It is important that your client be told at this stage what the product is likely to cost. You usually express this amount as the total cost to complete the product, but you may also find it advantageous to point out the cost of the requirements effort, or the costs of individual requirements.

Whatever you do, do not leave the costs in the lap of hysterical optimism. Make sure that this section includes meaningful numbers based on tangible deliverables.

41 Waiting Room

Requirements that will not be part of the next release. These requirements might be included in future releases of the product.

Content

Any type of requirement.

Motivation

To allow requirements to be gathered, even though they cannot be part of the current development. To ensure that good ideas are not lost.

Considerations

The requirements-gathering process often throws up requirements that are beyond the sophistication of, or time allowed for, the current release of the product. This section holds these requirements in waiting. The intention is to avoid stifling the creativity of your users and clients, by using a repository to retain future requirements. You are also managing expectations by making it clear that you take these requirements seriously, although they will not be part of the agreed-upon product.

Many people use the waiting room as a way of planning future versions of the product. Each requirement in the waiting room is tagged with its intended version number. As a requirement progresses closer to implementation, then you can spend more time on it and add details such as the cost and benefit attached to that requirement.

You might also prioritize the contents of your waiting room. “Low-hanging fruit”—requirements that provide a high benefit at a low cost of implementation—are the highest-ranking candidates for the next release. You would also give a high waiting room rank to requirements for which there is a pent-up demand.

42 Ideas for Solutions

When you gather requirements, you focus on finding out what the real requirements are and try to avoid coming up with solutions. However, when creative people start to think about a problem, they always generate ideas about potential solutions. This section of the template is a place to put those ideas so that you do not forget them and so that you can separate them from the real business requirements.

Content

Any idea for a solution that you think is worth keeping for future consideration. This can take the form of rough notes, sketches, pointers to other documents, pointers to people, pointers to existing products, and so on. The aim is to capture, with the least amount of effort, an idea that you can return to later.

Motivation

To make sure that good ideas are not lost. To help you separate requirements from solutions.

Considerations

While you are gathering requirements, you will inevitably have solution ideas; this section offers a way to capture them. Bear in mind that this section will not necessarily be included in every document that you publish.

43 Project Retrospective

Content

At the end of every project you should reflect upon what methods were used that worked out well and should be repeated in the future, and also what methods did not work out well and should be avoided. Any recommendations, suggestions, or ideas for how to do things better in the future should also be documented

Motivation

To learn from experience, and to continually strive for process improvement.

Considerations

When things don't go well, it is important to distinguish whether the methods themselves were poor, or simply poorly implemented in this particular case, or whether they just weren't right for this particular project / group of engineers.

VI Glossary

The glossary defines terms that may not be familiar to all readers. This is especially important if the document is expected to reach a wide and varied audience, such as school children. The glossary may be placed at either the beginning or the end of the document.

Flotsam: Any part of a ship or its cargo found floating on the water, whether it was deliberately or accidentally lost by its original owners.

Jetsam: Any part of a ship or its cargo that is deliberately cast off (jettisoned) by its original owners, generally in order to lighten the ship, whether it floats or sinks.

VII 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.

VIII 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. }