

Dungeon-Crafter



Create or Explore User-made Adventures

Prepared by
Group 12: Jay Dave, Dominick Rauba,
Vincent Orea, Garret Felker
University of Illinois Chicago

Fall 2014

[1][2][3]

Table of Contents

	List of Figures.....	7
	List of Tables.....	9
I	Project Description	10
1	Project Overview	10
2	The Purpose of the Project	10
2a	The User Business or Background of the Project Effort.....	10
2b	Goals of the Project.....	10
2c	Measurement.....	11
3	The Scope of the Work.....	11
3a	The Current Situation.....	11
3b	The Context of the Work.....	11
3c	Work Partitioning.....	12
3d	Competing Products	12
4	The Scope of the Product	12
4a	Scenario Diagram.....	12
4b	Product Scenario List	13
4c	Individual Product Scenarios	13
5	Stakeholders	13
5a	The Client.....	13
5b	The Customer	13
5c	Hands-On Users of the Product	13
5d	Priorities Assigned to Users	14
5e	User Participation.....	14
5f	Maintenance Users and Service Technicians.....	14
5g	Other Stakeholders	14
6	Mandated Constraints.....	14
6a	Solution Constraints.....	14
6b	Implementation Environment of the Current System	15
6c	Partner or Collaborative Applications	15
6d	Off-the-Shelf Software.....	16
6e	Anticipated Workplace Environment	17
6f	Schedule Constraints.....	17
6g	Budget Constraints	18
7	Naming Conventions and Definitions	18
7a	Definitions of Key Terms	18

7b	UML and Other Notation Used in This Document	19
7c	Data Dictionary for Any Included Models	19
8	Relevant Facts and Assumptions.....	19
8a	Facts	19
8b	Assumptions	19
II	Requirements	19
9	Product Use Cases	19
9a	Use Case Diagrams	20
9b	Individual Product Use Cases	20
10	Functional Requirements.....	23
11	Data Requirements	24
12	Performance Requirements	24
12a	Speed and Latency Requirements	24
12b	Precision or Accuracy Requirements	25
12c	Capacity Requirements	25
13	Dependability Requirements	25
13a	Reliability Requirements.....	25
13b	Availability Requirements.....	25
13c	Robustness or Fault-Tolerance Requirements.....	25
13d	Safety-Critical Requirements	26
14	Maintainability and Supportability Requirements.....	26
14a	Maintenance Requirements	26
14b	Supportability Requirements.....	26
14c	Adaptability Requirements.....	26
14d	Scalability or Extensibility Requirements.....	26
14e	Longevity Requirements	26
15	Security Requirements.....	26
15a	Access Requirements	26
15b	Integrity Requirements.....	27
15c	Privacy Requirements	28
15d	Audit Requirements.....	28
15e	Immunity Requirements.....	28
16	Usability and Humanity Requirements	29
16a	Ease of Use Requirements.....	29
16b	Personalization and Internationalization Requirements.....	29

16c	Learning Requirements	29
16d	Understandability and Politeness Requirements	29
16e	Accessibility Requirements	29
16f	User Documentation Requirements	30
16g	Training Requirements	30
17	Look and Feel Requirements	30
17a	Appearance Requirements	30
17b	Style Requirements	30
18	Operational and Environmental Requirements	30
18a	Expected Physical Environment	30
18b	Requirements for Interfacing with Adjacent Systems	31
18c	Productization Requirements	31
18d	Release Requirements	31
19	Cultural and Political Requirements	31
19a	Cultural Requirements	31
19b	Political Requirements	31
20	Legal Requirements	32
20a	Compliance Requirements	32
20b	Standards Requirements	32
III	Design	32
21	System Design	32
21a	Design goals	32
22	Current Software Architecture	34
23	Proposed Software Architecture	34
23a	Overview	34
23b	Class Diagrams	35
23c	Dynamic Model	35
23d	Subsystem Decomposition	37
23e	Hardware / software mapping	39
23f	Data Dictionary	39
23g	Persistent Data management	40
23h	Access control and security	40
23i	Global software control	41
23j	Boundary conditions	42
24	Subsystem services	42
25	User Interface	43

26	Object Design	47
26a	Object Design trade-offs	47
26b	Interface Documentation guidelines.....	48
26c	Packages.....	49
26d	Class Interfaces	50
IV	Test Plans.....	56
27	Features to be tested / Not to be tested	56
28	Pass/Fail Criteria	57
29	Approach	58
30	Suspension and resumption	58
31	Testing materials (hardware / software requirements)	60
32	Test cases.....	60
33	Testing schedule	64
V	Project Issues	65
34	Open Issues.....	65
35	Off-the-Shelf Solutions	65
35a	Ready-Made Products	65
35b	Reusable Components.....	66
35c	Products That Can Be Copied	66
36	New Problems	67
36a	Effects on the Current Environment.....	67
36b	Effects on the Installed Systems.....	67
36c	Potential User Problems.....	67
36d	Limitations in the Anticipated Implementation Environment That May Inhibit the New Product.....	67
36e	Follow-Up Problems	68
37	Tasks.....	68
37a	Project Planning	68
37b	Planning of the Development Phases	68
38	Migration to the New Product	69
38a	Requirements for Migration to the New Product.....	69
38b	Data That Has to Be Modified or Translated for the New System	69

39	Risks	70
40	Costs	70
41	Waiting Room	70
42	Ideas for Solutions.....	71
43	Project Retrospective.....	71
VI	References / Bibliography	72

List of Figures

Figure 1 - Scenario Diagram.....	12
Figure 2 - Implementation Environment of the Current System	15
Figure 3 - 3 rd Party Character Generator.....	16
Figure 4 – Use Case Diagram	20
Figure 5 – Data Requirement Diagram.....	24
Figure 6 – Proposed Architecture Diagram	34
Figure 7 – Class Diagram	35
Figure 8 – Dynamic Model for Login/Signup Use Case	35
Figure 9 – Dynamic Model for Group Formation and GamePlayManagement Use Case	36
Figure 10 – Dynamic Model for DisplayingStats.....	36
Figure 11 – Dynamic Model for UpdatingStats.....	36
Figure 12 – Dynamic Model for Complete/Exit Adventure use case.....	36
Figure 13 – Subsystem Services Diagram	37
Figure 14 – Hardware/software mapping Diagram	39
Figure 15 – Persistent Data management Diagram.....	40
Figure 16 – Subsystem Services Diagram	43
Figure 17 – Login/Signup User Interface Design.....	44
Figure 18 – Lobby User Interface Design.....	44
Figure 19 – Group Formation User Interface Design	45
Figure 20 – Adventure User Interface Design	46
Figure 21 – Show Statistics User Interface Design	46
Figure 22 – Package Design	49
Figure 23 – Timeline of major HTML changes.....	65
Figure 24 – Statistics of visitors using WebGL.....	66

Figure 25 – Percentage of Internet-enabled PCs having various player	66
Figure 26 – Agile Life Cycle Model.....	68

List of Tables

Table 1- Subsystem Decomposition	38
Table 2- Data Dictionary	40
Table 3- Access control Matrix.....	41

I Project Description

1 Project Overview

Dungeon crafter is a web based multiplayer game where you can create a character to play in an adventure consisting of various levels. Game play involves the user to form a group and play against enemies who will appear at each level.

2 The Purpose of the Project

2a The User Business or Background of the Project Effort

Table top games such as Dungeons and Dragons have been popular modes of entertainment for decades. However, there have not been any viable online alternatives that accurately emulate the classic game. Business/Client i.e. ***Forgotten Realms*** wants to explore this opportunity to implement Dungeons and Dragons using a web based application to eliminate the difficulties faced in the classic table top game.

In the original dungeons and dragons paper-based game (not Dungeon Crafter) keeping track of game information can be daunting if the group meets only a few times per month. This can be solved by a simple creating a web based adaption (Dungeon Crafter). Not only can the internet solve the issue of record keeping, but it also allows users to play together from anywhere at any time most convenient for it's' players. Meeting in person can be difficult to fit multiple schedules in the original paper-based game. Dungeon Crafter allows users to play without being together 'in person'.

2b Goals of the Project

Our goal is to create a web based role playing game that gives the feel of a classic table top role playing game, where groups of 5 players set off on premade adventures from the comfort of their respective houses.

Player will be assigned a character when the signup for the game with default skill set. As the players defeat an opponent, they will advance their skill set.

Once the group of players who has set off for an adventure completes that adventure, they are displayed a statistics page that displays their current skill set as well as the total gain during the adventure.

2c Measurement

A goal of this project is to give adventure seekers a platform to allow their creativity to be displayed in the form of adventures and scenarios, to enjoy by numerous players and enthusiasts. Having an active player based community will allow hundreds if not thousands of different adventures created by the *Forgotten Realms* for the community. The bigger and more active the community is, the easier it is to measure this projects overall impact in online role-playing games. We believe that having a large player community will spur the creation of many unique and entertaining adventures. The goal of the project is to create at least 50 adventures each consisting of 100 levels wherein the player will face different enemies.

3 The Scope of the Work

3a The Current Situation

Currently this style of game is confined to paper and pencil, and a hosts' location, at a very specific time. These events are inconvenient to attend, as it is difficult to fit into the schedule of several people. People are also restricted to playing with people they know personally. Players also need to purchase various rule books and dice.

3b The Context of the Work

The product won't be affected by any external systems, though the rules that will be followed by our system are as follow:

1. Game includes formation of groups of online players.
2. Once the groups are formed, the game play begins; Different levels (1-100), each level has 2 enemies.
3. Any one player (as per the turn) attacks enemy using the attack button on his screen.
4. If skill set of Enemy is more than skill set of Player
Then, Player loses his energy by 1 and if his energy has been exhausted then player loses and he gets eliminated from the adventure, his stats are shown to him. He may exit the application from there.
5. If skill set of Player is more than or equal to skill set of Enemy
Then, player gains skill set by 3%
AND
Player gains experience by 5%
AND
Get to next enemy OR level.
6. If Level count of adventure >100
Then, display: 'Adventure Completed', show player statistics. He may exit the application from there.

3c Work Partitioning

Our product isn't affected by any external system/game. Thus there won't be any work partition for our product.

3d Competing Products

The original Dungeons & Dragons table top game (or any related role-playing game).

4 The Scope of the Product

The scope of our product will be a utility for people to form a group and participate in an adventure. Once an adventure is initialized with a group the set of rules that have been predefined by the product will dictate the options that they have for their turns.

4a Scenario Diagram

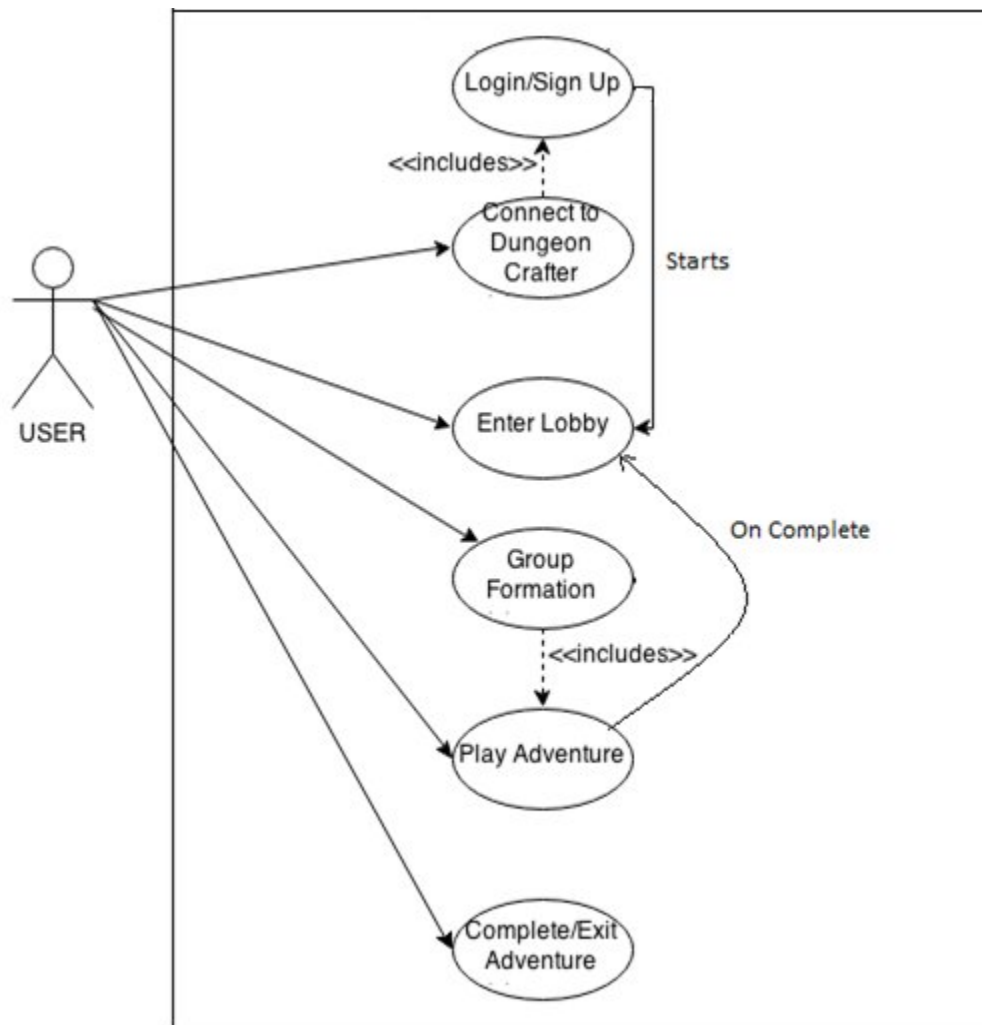


Figure 1 - Scenario Diagram

4b Product Scenario List

- Connect to Dungeon Crafter URL
- Login/Signup
- Lobby
 - Group Formation: Player looks for group
 - Log out: Player exits lobby
- Player completes game and enters statistic lobby

4c Individual Product Scenarios

Connect to Dungeon Crafter: User navigates to the Dungeon Crafter URL to play the game.

Login/Signup: In case of existing user, user is prompted for User_id and Password. In case of new user, user is prompted for user details and a new account is setup.

Lobby: User can start playing game by clicking on Group Formation button which forms a group of player post which an adventure is started. Also, the user can log out from the application from the lobby.

Statistics: Once the player completes an adventure, he/she is displayed his/her statistics

5 Stakeholders

5a The Client

John Mathias, Marketing head of Forgotten Realms has requested this project so that world can have an online role playing game and thereby add a new dimension to their firms' services and products.

5b The Customer

Board game enthusiasts who are interested in playing turn based, role playing games online would be our largest customer.

5c Hands-On Users of the Product

There won't be any hands-on users of the product other than the customers who are looking to eliminate the barriers of acquaintance and place to play the legacy board game.

5d Priorities Assigned to Users

- Key users: Players who will be forming a group and start the adventure will be our key users as they will drive our game by experiencing different inbuilt adventures.

5e User Participation

Game players are the largest participants as they are the ones who will experience the adventure there by advancing their statistics. More the adventures played, more will be the player's statistics.

5f Maintenance Users and Service Technicians

The server administrators will be considered as the maintenance user and service technician as they will be responsible to update the versions as well as apply required patches.

5g Other Stakeholders

- Sponsors

Forgotten Realms will be the sole sponsor for this project. Since that is the case, they will give the seal of approval, or have a large influence in the course of our product development.

- Testers (game players)

The development team will be testing the application followed by Forgotten Realms' testing team before first release. A beta version will be available on our website for open testing. Any bugs can be forwarded to our development team.

6 Mandated Constraints

This section describes constraints on the eventual design of the product. They are the same as other requirements except that constraints are mandated.

6a Solution Constraints

Description: The product shall operate through website using any kind of system.

Rationale: The client has multiple potential users' through-out the world that will access the application through client's website using their system.

Fit criterion: The product shall be approved by the Forgotten Realms' testing team.

Description: The product shall provide an 'exit' option throughout the application windows, so that user can exit the application on their discretion.

Rationale: The client customers would like to exit the application at any point of time during the play.

Fit criterion: The product shall be approved by the Forgotten Realms' testing team.

Description: The product shall have a feature of time-out after 2 minutes while 'looking up for group'.

Rationale: Client doesn't want its customers to wait for a long period of time.

f The product shall have a feature of self time-out after 2 minutes. The product shall be approved by the Forgotten Realms' testing team.

6b Implementation Environment of the Current System

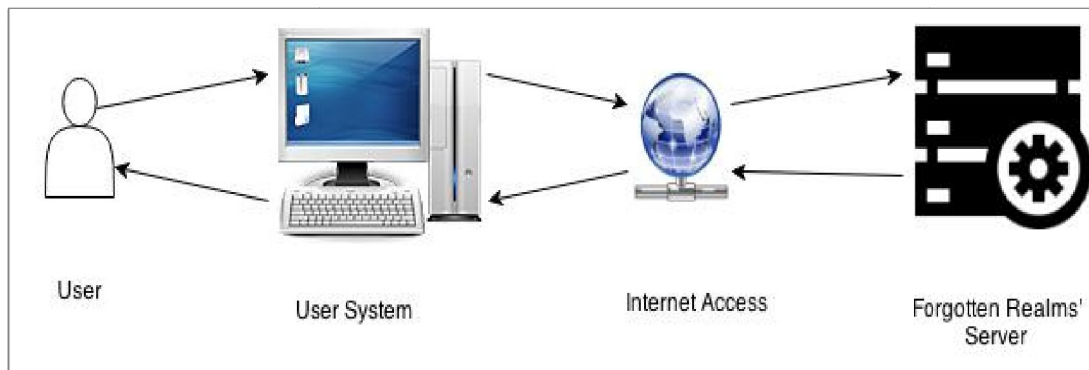


Figure 2 - Implementation Environment of the Current System

6c Partner or Collaborative Applications

Third-party applications such as various Skin Modifier applications can be used by Game Creator, at their discretion.

Motivation

Allowing users to import items or character skins from 3rd party applications offers scenarios a chance to increase uniqueness among other scenarios and offers a platform for creative minds to showcase their work to others.

Third Edition (3.5) Dungeons and Dragons Character Generator

Version -- 4.9

This character generator is is used by [Turn Watcher](#)

About this Generator

FAQs

Check for Updates

Display Empty Sheet

Display Help

Name:

Player:

Race:

Starting class:

Figure 3 - 3rd Party Character Generator

Figure 3 displays the beginning of a 3rd party generator (accessible by anyone) process that will allow a user to build their own character to be used in scenarios. The user must select traits and abilities unique to their 'race' and 'class', which will enable them to perform certain tasks better than others. The file created from this application can then be imported as long as the file types match and there is no scenario harming code or malware attached.

Considerations

3rd party applications must be able to export their files to a compatible file type that will function correctly on our servers.

6d Off-the-Shelf Software

Content

Off-the-Shelf requirements:

- Compatible file types (item sets, landscapes, characters)
- Easy to use interface
- NO malware or malicious file(s) can be used.

Motivation

Physical implementations of randomization could increase user physical participation with their scenarios

Considerations

Any copyrighted material must have proper permissions before being used in scenarios and items and characters resembling those of other's ideas and work will be subject to rejection to our server.

6e Anticipated Workplace Environment

Content

User's workplace environment will be entirely up to the User's preferences. Ideas of workplaces include but are not limited to, private home(s), public places such as coffee shops, libraries, and video arcades; granted there is an available internet connection. Depending on the location chosen by the User comfort and noise levels may vary.

This describes the workplace in which the users are to work and use the product. It should describe any features of the workplace that could have an effect on the design of the product, and the social and culture of the workplace.

Motivation

To allow Users the opportunity to play scenarios in locations they feel most comfortable.

Considerations

Poor Internet connection(s) will limit the User's abilities to participate scenarios properly.

6f Schedule Constraints

Content

3 am (GMT) to 4 am (GMT) weekdays, server maintenance.

Motivation

To perform server modifications including but not limited to, server upgrades, game improvements, enhanced security implementations.

Examples

Ensure servers are running properly and to administer updates to gameplay and other features dealing with the server.

Windows of marketing opportunity.

Scheduled maintenance allows the chance for advertisements to display on webpage during server service hours.

Considerations

- What happens if we don't build the product by the end of the calendar year?

Developers would have to perform scheduled server maintenance quicker and more efficiently to ensure user accessibility during non-service hours.

- What is the financial impact of not having the product by the beginning of the Christmas buying season?

Christmas themed advertisements to be displayed to users during service-hours. Which should increase sales of advertised products.

6g Budget Constraints

Content

Budget to maintain minimal server cost - \$4000.00/mo

Developer(s) salary - \$150000

Motivation

Est. 20000gb bandwidth/mo

Considerations

Yes, as user/fan base grows, marketing solutions such as advertising during maintenance hours can pay portion of server cost and developer compensation.

7 Naming Conventions and Definitions

7a Definitions of Key Terms

XP – (Experience) Refers to the progress made towards a character's next level.

IN – (Intelligence) Refers to one of the power of a player.

ST – (Strength) Refers to one of the power of a player.

7b UML and Other Notation Used in This Document

This document generally follows the Version 2.0 OMG UML standard, as described by Fowler in [4].

7c Data Dictionary for Any Included Models

We do not have any external model that has been included. Though there is an authority given to the user to use off the shelf software in a legitimate manner. The rules to be used are explained under ‘The context of the work section’ ([section 3b](#))

8 Relevant Facts and Assumptions

8a Facts

This application will combine the convenience of an internet-based game, with the joys of a classic table-top roleplaying game. The application will eliminate the need for the players to keep track of numerous values and information. However the gameplay rules will not be nearly as extensive and detailed as they are in the classic game. Only rules deemed necessary to fulfill the ‘classic board game experience’ will be implemented in the game. This is due to the enormity of the rule set of the table top game.

8b Assumptions

- The user will have access to the internet.
- Multiple users will be able to play simultaneously.
- Users will have a basic understanding of the rules used.
- Every adventure will have 100 levels
- Each level will have 2 enemies and the skill set of enemies increase as the level increases by a factor of 0.05.

II Requirements

9 Product Use Cases

This section begins to describe in more specific and precise detail exactly what steps the system takes in the course of its performance. Use cases serve not only to more specifically define the system (and its boundaries), but also to identify functional requirements, to identify initial objects / classes, and to organize the work.

9a Use Case Diagrams

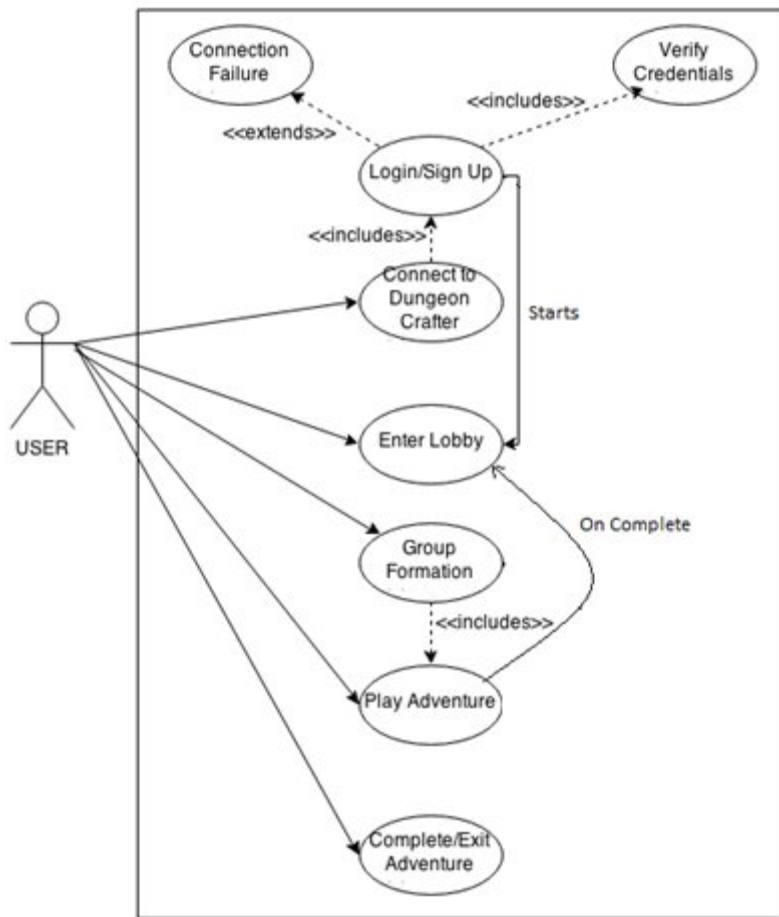


Figure 4 – Use Case Diagram

9b Individual Product Use Cases

Use Case Name	Login
Participating Actors	User
Flow of Events	1.) User enters login credentials 2.) Credentials validated by Server 3.) If logon success, user is brought to character selection screen 4.) If logon fail, remain on logon screen
Pre-Condition	Connect to Dungeon Crafter

Post-Condition	Enter Lobby OR Connection Failure
Quality Requirements	Login gets authenticated within 5 seconds.

Use Case Name	Sign Up
Participating Actors	User
Flow of Events	1.) User enters his details like email id, password, Name and age 2.) Email id validated by Server 3.) If signup success, a confirmation email is sent to user 4.) User may log in after clicking link in confirmation email
Pre-Condition	Launch game client
Post-Condition	Confirmation email sent.
Quality Requirements	Confirmation email gets sent within 5 minutes.

Use Case Name	Group Formation
Participating Actors	User
Flow of Events	1.) User is clicks on Group formation button on Lobby 2.) User is randomly matched with other users.
Pre-Condition	User selects “Look For Group” from the Lobby
Post-Condition	User selects “Exit”, OR the adventure begins.

Quality Requirements	Game spends a maximum of 2 minute searching for a group for the user. After 2 minute, the game displays a message telling the user a group could not be found, and to try again later.
----------------------	--

Use Case Name	Play Adventure
Participating Actors	User
Flow of Events	<ol style="list-style-type: none"> 1.) User clicks attack button 2.) Existing characters are loaded from the User Database 3.) Game play rules are applied and result is updated
Pre-Condition	All users select “Ready” in the “Look For Group” menu.
Post-Condition	Users complete or fail the adventure.
Quality Requirements	State of adventure is consistent for all participating users.

Use Case Name	Complete/Exit Adventure
Participating Actors	User
Flow of Events	<ol style="list-style-type: none"> 1.) Use clicks exit button 2.) The product exits the adventure and the game as well.
Pre-Condition	User defeats the enemies in the final room of the dungeon.
Post-Condition	All surviving users are taken to the “Adventure Statistics” screen
Quality Requirements	Statistics displayed on the “Adventure Statistics” are accurate to what actually happened in the adventure.

10 Functional Requirements

1. The adventure database must be able to store the adventure data file, with the primary key being the name of the adventure.
2. The application must generate a data file that represents the current state of each character.
3. The user database must be able to store the character data file, with the primary key being the user who created it.
4. The application must remove inactive users from the 'Looking for group' list.
5. The application must be able to display graphics of the adventure by reading the adventure data file.
6. The adventure data file must be dynamically updated based on the actions of the players in it.
7. Change in character position, or environment variables must be immediately reflected on the users' graphical interface.
8. All adventures must have a consistent initial state, and a condition that determines whether or not the adventure is complete.
9. An adventure is automatically terminated when either all the characters in the group have died, or when the adventure is completed.
10. A user may only level-up a character when he/she is NOT currently in an adventure.

11 Data Requirements

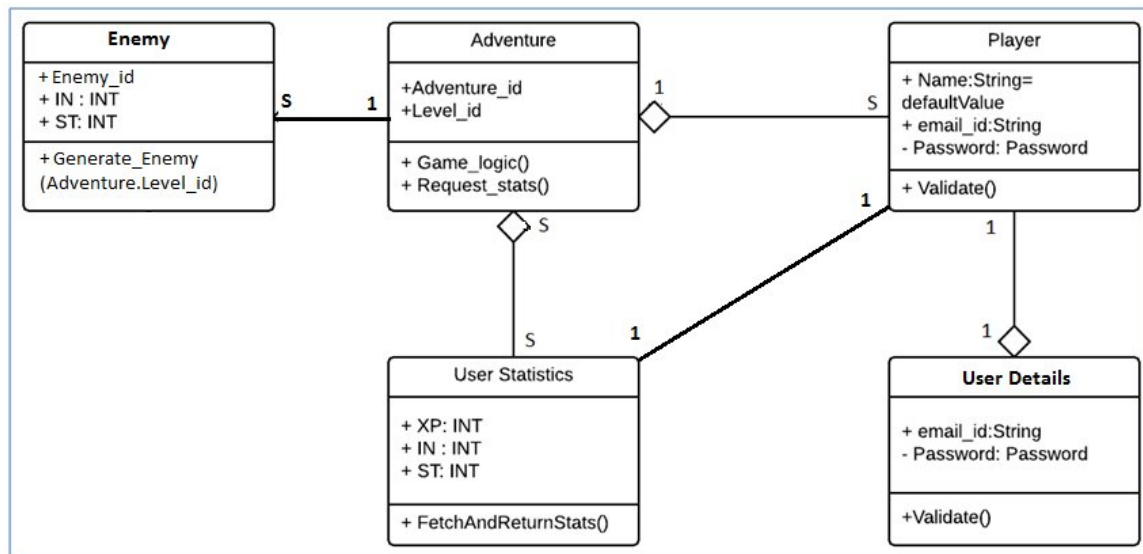


Figure 5 – Data Requirement Diagram

12 Performance Requirements

12a Speed and Latency Requirements

The product shall read/search from the database in no more than 5 seconds.

The product shall update the database in no more than 5 seconds.

No new profile processing shall take more than 5 seconds.

The product shall download the new environment change within 10 seconds of a change.

Overall, the application must feel responsive.

Fit Criterion

If any request takes longer than 5 seconds, the application must display a logo to indicate that the application is still processing the request. (Similar to the logo below)



12b Precision or Accuracy Requirements

The product should have a higher priority on speed than accuracy; however, each user statistics should have a scale of 7.

12c Capacity Requirements

The product shall cater for 500 simultaneous users within the period from 5:00 pm (GMT) to 1:00 am (GMT). Maximum loading at other periods will be 300 simultaneous users.

During a launch period, the product shall cater for a maximum of 100 users to be in the inner chamber.

13 Dependability Requirements

13a Reliability Requirements

The product shall not fail more than once per day. In case of failure, it will have a fail-safe recovery mechanism to allow the product to finalize any transactions that were left incomplete.

No data shall be lost or damaged in the database during an update whenever User is offline or suddenly becomes offline while an update is in progress.

No data shall be lost or damaged while saving user preferences and the system suddenly loses power and shuts off.

No data shall be lost or damaged in the event of any other failure.

All database INSERTs and UPDATEs should be done as a TRANSACTION to ensure ACID (Atomicity, Consistency, Isolation, and Durability) compliance.

13b Availability Requirements

The product shall be available for use 24 hours per day, 365 days per year.

The product shall achieve 99 percent uptime.

Eventually, the product might be ported to mobile devices in which case the users will need to be able to download the product at anytime from anywhere

13c Robustness or Fault-Tolerance Requirements

In case of errors during database insertion and/or update queries, the application shall save the queries to a file and notify the owner of the problem. Whenever the database comes back online, all the saved queries shall be executed.

13d Safety-Critical Requirements

There are no safety concerns for this product.

14 Maintainability and Supportability Requirements

14a Maintenance Requirements

The product version updates should take place between 3 am (GMT) to 4 am (GMT) and should not take more than 5 minutes.

The product must not face a server down time of more than 15 minutes.

14b Supportability Requirements

The product shall refer the user to an online manual that shows the user how to use the product, start an adventure, and enter an adventure and most importantly how to create adventures.

14c Adaptability Requirements

The product is expected to be platform independent.

The product shall run on any browser that support flash player.

14d Scalability or Extensibility Requirements

The product shall be capable of processing the existing 500 customers. This number is expected to grow to 1500 customers within three years.

The product shall be able to process 1000 transactions per hour within two years of its launch.

14e Longevity Requirements

The product is expected to operate without significant performance degradation for a minimum of five years.

The product shall be expected to operate within the maximum maintenance budget for a minimum of five years.

15 Security Requirements

15a Access Requirements

The material in this section will cover the different aspects of access to Dungeon Crafter. The motivation of this section is to describe (in detail) how and why people will be granted certain levels of access in Dungeon Crafter.

The database administrator will have access to all of the data stored on the servers.

All general end users will have read access to their character.

This product does not contain highly sensitive information.

Only administrator should have write access to statistics database in other words, no player can edit his/her statistics.

Only one login is allowed per player i.e. one player cannot login multiple times.

15b Integrity Requirements

This section will cover the moral code of our data.

Users can report players for inappropriate behavior. When a player gets reported a moderator who manages good player ethics can analyze chat logs to determine if a player is deserving of punishment (such as being banned from the game). Our product will ensure that all players are treated well by one another.

Our product will also ensure that scenarios are appropriate and follow our guidelines which will be covered in another section.

We will require a DBA guru and a strong networking team to be able to backup and maintain our data. This will prevent unexpected catastrophes from shutting down our servers and game.

A secure and remote data center would be the best place to place our servers. We will strive to offer an honest and open product to meet the acceptance of the general public's requirements.

The product shall prevent incorrect data from being introduced.

The product shall protect itself from intentional abuse.

The product shall prevent multiple logins of same account (player).

The product shall update the player database with appropriate statistics once the adventure is complete.

The product shall treat multiple related queries as a transaction i.e. all of them should get executed successfully or none.

15c Privacy Requirements

The product will ensure that all laws related to privacy of an individual's data are observed, to ensure that the product complies with the law, and to protect the individual privacy of your customers.

Few people today look kindly on organizations that do not observe their privacy. The product shall make its users aware of its information practices before collecting data from them.

The product shall notify customers of changes to its information policy.

The product shall reveal private information only in compliance with the organization's information policy.

The product shall protect private information in accordance with the relevant privacy laws and the organization's information policy.

Customers must always be in a position to give or withhold consent when their private data is collected or stored. The most sensitive information that is held would be the email addresses of authenticated users.

15d Audit Requirements

Our product will authenticate the login credentials (email id and password) upon signing in to verify that the proper user is logging into their proper account.

We will build a system that complies with the appropriate audit rules.

The product shall make sure that the above three requirements are fulfilled.

The fulfillment of above three product requirement assures that the audit requirements are met.

15e Immunity Requirements

Our product will not have software installed on the end-user PC. This game will be completely browser based and will require safely written code to prevent any injections into our database code.

The product will be secure and safe from malicious interference.

Developers of our product will require good programming practice and will have to create 'safe' code to prevent database injections.

The product should not allow any user(s) other than the administrator(s) or designer(s) to change any interface that can compromise user data or make user prone to virus attacks.

16 Usability and Humanity Requirements

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

16a Ease of Use Requirements

It will be the users right to enjoy having a simple to use interface.

FAQ's will be available along with a forum.

When our product is initialized by the user, they will have a multitude of options to select from:

- Group Formation
- Exit

We expect users who are ages 13-45 to learn how to use our game almost immediately.

Fit Criteria:

This will require testing: 65% of new users/ tester to learn how to use our product within 1 hour.

16b Personalization and Internationalization Requirements

Our initial supported language will be English which is most commonly spoken amongst videogame enthusiasts worldwide. It will be our mission to make this game as accessible as possible. The larger our community, the more successful our game will be. We will have to abide by local laws (which shouldn't be an issue and is discussed in more detail within the legal section).

16c Learning Requirements

Any users that do not know how to use our product will be able to navigate to the "help" section on our website where they can watch tutorials read FAQ's visit the forums for help.

16d Understandability and Politeness Requirements

Our product will have a reporting system where users who feel they have been wronged by others can submit tickets to moderators.

Our product will not tolerate bullying within our community.

It is the right of our end users to be able to participate in an open and accepting community regardless of gender, ethnicity and regional origin.

16e Accessibility Requirements

Our product will be accessible to anyone who can operate a computer and has an internet connection.

16f User Documentation Requirements

Documentation that will be provided:

- Terms of service
- Anti bullying agreement
 - An agreement that will result in a permanent ban if not honored
- Downloadable
 - All documentation is available online and downloadable
- Tutorial
 - Game guide

Legal team will be required to keep terms of service and anti bullying agreement up to date.

16g Training Requirements

Training will be offered within the documentation. Additionally, Tutorials will also be available on our forum.

17 Look and Feel Requirements

17a Appearance Requirements

The appearance will be relatively simple as compared to other web based multi-player games.

The spirit of this product will be dark and ominous.

The background will be brimstone and is intended to elicit feelings of fear.

Our product shall cater to those who like the thought of evil mythical creatures and vanquishing them.

At the top of the web interface “Dungeon Crafter” will be in bold red at the top.

17b Style Requirements

The product shall appear bold and powerful.

The product shall have elements of darkness and elements of light. The darkness and light will be battling for control of the web interface.

18 Operational and Environmental Requirements

18a Expected Physical Environment

Our product can be operated anywhere a computer can function and an internet connection is available.

18b Requirements for Interfacing with Adjacent Systems

The Product shall require an internet connection.

The Product shall require an up-to-date web browser

- Our product will prompt users to update their web browser if behind more than 4 releases.

-

The Product shall require servers:

- Web server
- Database

The products data medium is the internet

18c Productization Requirements

The product will require an up-to-date web browser to launch the game.

The product will be able to operate on any computer with an internet connection regardless of the client machine's hardware specifications.

18d Release Requirements

Server maintenance will happen once per week, between 3 am (GMT) to 4 am (GMT) (this is an expected time that will have the minimum number of users). Server maintenance includes patches, server upgrades, game updates, interface updates, etc.

Server maintenance will be a regular occurrence to prevent our product from being down for extended periods of time.

19 Cultural and Political Requirements

19a Cultural Requirements

The product shall be developed giving careful thought to religious and ethnic groups, so as not to be offensive or disrespectful.

The product shall not be able to identify the population of specific races in a particular area.

19b Political Requirements

There are no political requirements for our product.

20 Legal Requirements

20a Compliance Requirements

Personal information shall be implemented so as to comply with the Data Protection Act.

All the adventures displayed in the lobby must be gotten from adventure database to which it is interfaced.

The product shall adhere to all the copyright rules of the rule guide that the product shall use.

The product shall adhere to all the copyright of the plug-in software that are being used.

20b Standards Requirements

The product can use VAP methodology to create adventures.

III Design

21 System Design

21a Design goals

Performance Criteria:

1. *Response time:* The product should acknowledge the user request within 5 seconds of the request so that the user understands that the system has received user request and is processing the request.
2. *Throughput and Memory:* The product shall be capable of processing the existing 500 customers. This number is expected to grow to 1500 customers within three years. The product shall also be able to process 1000 transactions per hour within two years of its launch.
3. *Accuracy:* Since our product need not be highly accurate with the statistics and other game data, however, a precision for each user statistics of a scale 7 is expected.
4. *Speed:* The product (being a web based multi-player game) must place a high priority on speed than accuracy, and so the physics engine for a computer game may make some rough approximations and assumptions that allow it to run as fast as possible while sacrificing accuracy. The product must feel responsive, as such the time required to interact with database should be kept to a minimum - preferable between 2-5 seconds.

Dependability Criteria:

1. *Robustness/Fault Tolerance:* Since the system relies heavily on direct user input, care must be taken to make sure that any user input does not cause any runtime errors or crashes due to unsanitized input. Also, the database integrity must be kept in mind while dealing with database updates i.e. treats a set of associated queries as a single transaction.
2. *Reliability:* Since, the reliability of the product is dependent upon the availability of server as well as the database, a backup server as well as a redundant database backup plan is highly desirable. Server down time and database crashes must be minimized using required protocols like ACID property for database and resourcing and regular space cleanup activity at server.

Maintenance Criteria:

1. *Scalability or Extensibility:* The product design should be modular, reason being the extensibility of the product with respect to interactions with other players by adding a chat module or single player game play with a group of Artificial Intelligence team mates.
2. *Modifiability and Adaptability:* Designing the product in a modular manner will make the future modification of the product really easy. In addition, modular products are also easier to port to different application domains thus making the product more adaptable.

Cost Criteria:

1. *Development Cost:* A total development budget of US \$150000 is assigned.
2. *Maintenance Cost:* A maintenance budget of US \$4000/month is assigned.

End User Criteria:

1. *Usability:* The product should have a simple user interface that would consist for required buttons and each window should have the product icon and an exit button.

22 Current Software Architecture

There is no such game existing, so there is no current software architecture.

23 Proposed Software Architecture

23a Overview

Although the current architecture where the database acts like a server can fulfill the needs of the program, the design goals indicates that model/view/controller architecture will make more sense. In this mode, the view will be the user interface, where the player will give commands about the either the game play or the lobby options, the controller will control the commands to the database and rule logic, and the model will be the database management structure and it will be mostly handling the data.

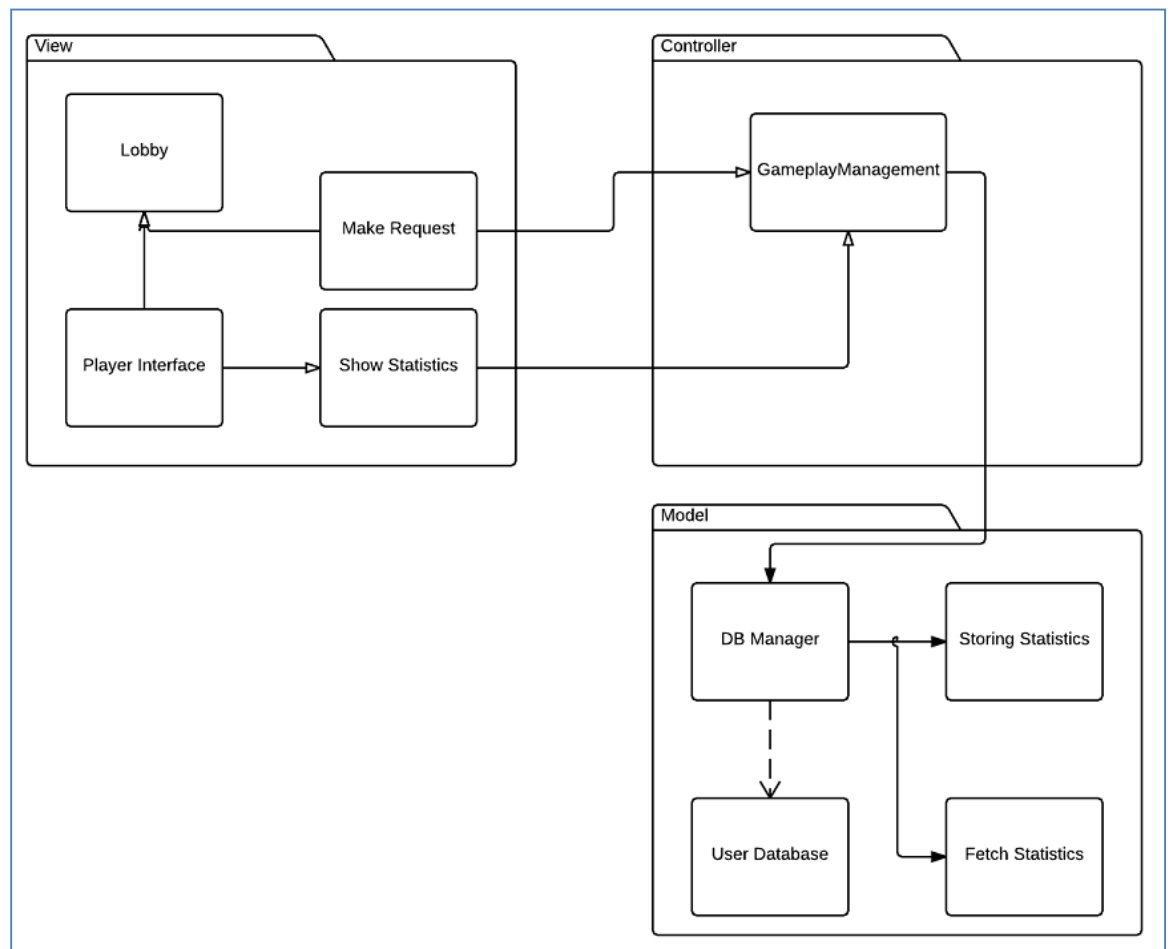


Figure 6 – Proposed Architecture Diagram

23b Class Diagrams

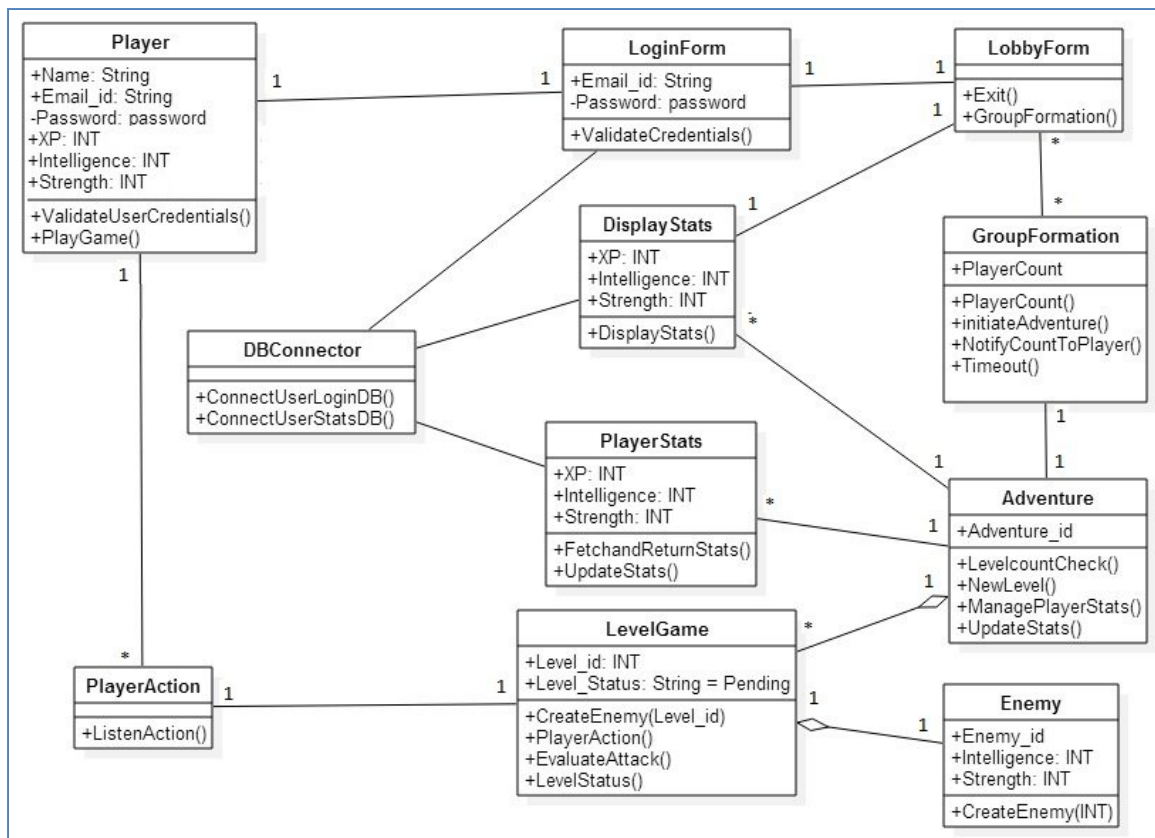


Figure 7 – Class Diagram

23c Dynamic Model

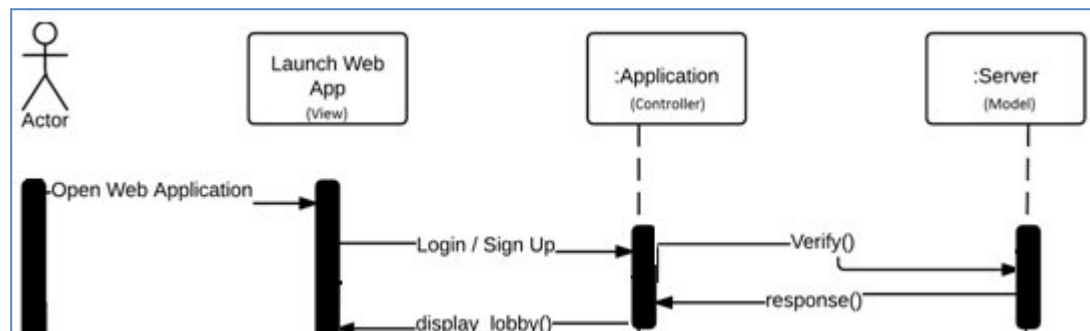


Figure 8 – Dynamic Model for Login/Signup Use Case

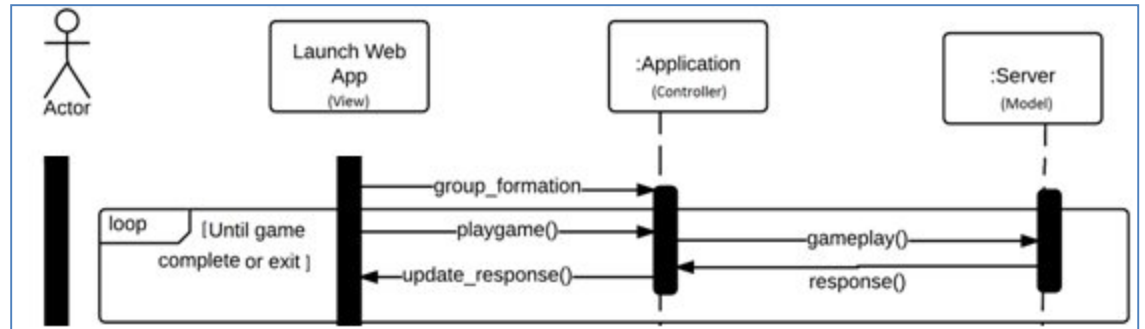


Figure 9 – Dynamic Model for Group Formation and GameplayManagement Use Case

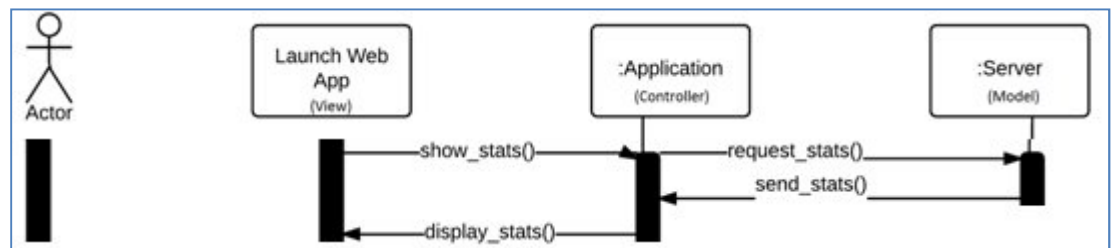


Figure 10 – Dynamic Model for DisplayingStats

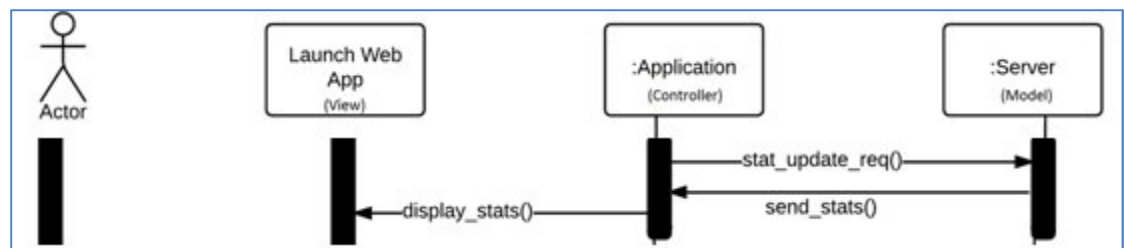


Figure 11 – Dynamic Model for UpdatingStats

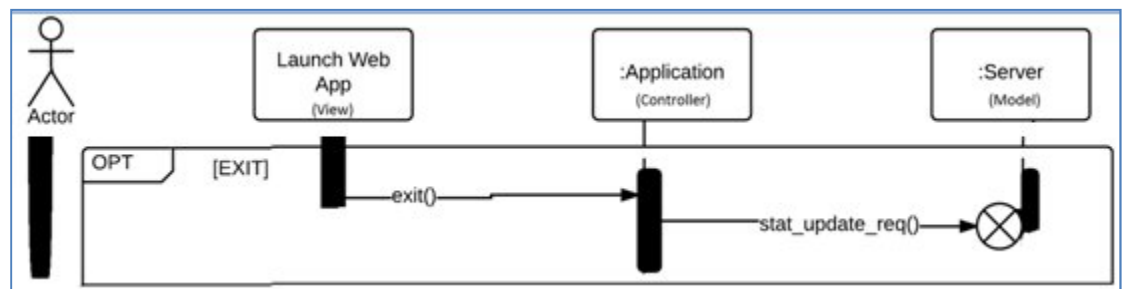


Figure 12 – Dynamic Model for Complete/Exit Adventure use case

23d Subsystem Decomposition

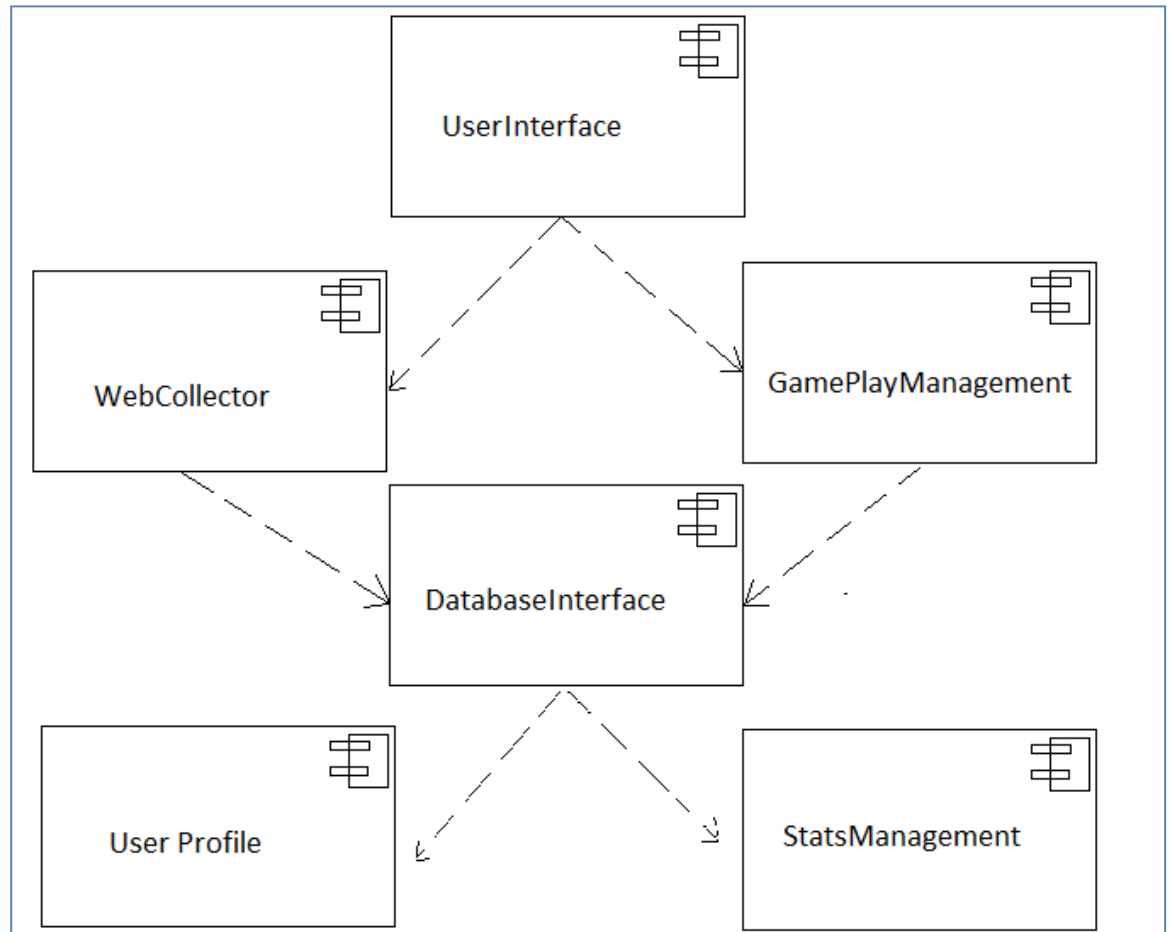


Figure 13 – Subsystem Decomposition Diagram

UserInterface	Handles display that is shown to User during use. Includes classes like LoginForm and LobbyForm.
---------------	--

User Profile	This subsystem handles sensitive information of the end user as they progress through adventures. This subsystem is also referenced for user verification during signup/login. This subsystem comprises of Player class.
GamePlayManagement	Handles the game logic including enemy statistics, possible interactions that can be made by the user during game play, keeps track of user progress during an adventure. This subsystem comprises of Group formation, Adventure, LevelGame and Enemy class.
DatabaseInterface	Generates queries that are sent to one of the databases to either store or update user statistics or verify and manage user validity during login or sign up. This subsystem comprises of databases and DBCollector class.
StatsManagement	This subsystem manages the stats (statistics) of each player like experience (XP), Intelligence (IN) and Strength (ST). It helps in updating, fetching and displaying the stats for each user. This subsystem comprises of DisplayStats and PlayerStats class.
WebCollector	Bookkeeping when a User signs up for the first time to manage and keep the User DB updated. Also, helps user to interact during the game play using action button.

Table 1- Subsystem Decomposition

23e Hardware / software mapping

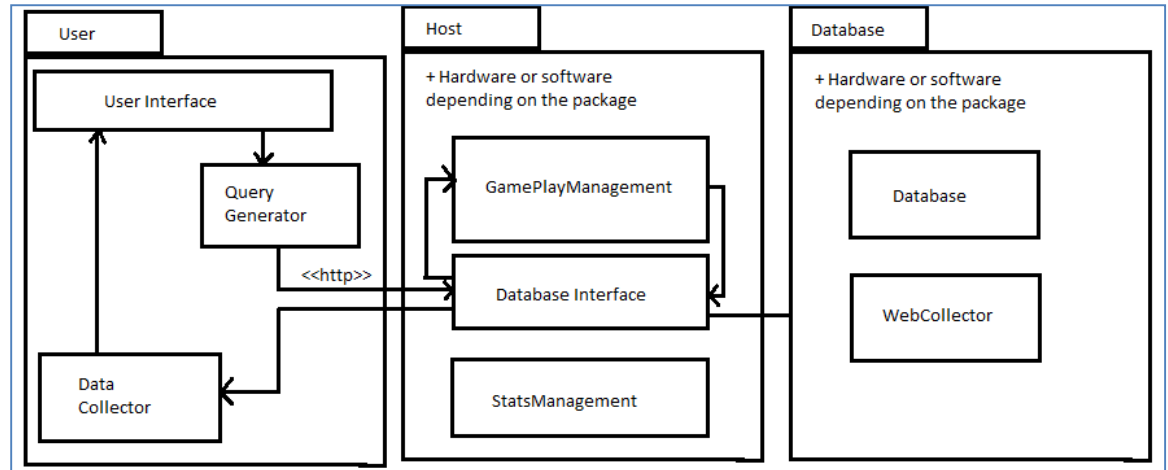


Figure 14 – Hardware/software mapping Diagram

23f Data Dictionary

Field Name	Data Type	Other Information
Email_id	String , Field Size = 15	Email id of the user
CharacterName	String, Field Size = 15	Name of the user
Password	Password, Field Size = 10	Password of the user
Skills	Int	Strength (ST), Intelligence (IN) These are the skills the player/Enemy uses to fight with the enemy. Initially for player, each will be 100.
Experience (XP)	Int	Experience (XP) Initially, assigned as 10
PlayerCount	Int	Keeps count of the number of players in the group.
Adventure_id	Int	Stores the adventure id so that a unique id can be refer to an adventure taking place.
Level_id	Int	Stores the level id of the adventure so that level threshold can be kept track of. Also, used to generate enemy . Value of this id increments after each level is completed.
Level_status	String, Field Size = 10	Default = 'Pending', determines if the level is completed by the group or not. On complete, this field is changed to 'complete'.

Enemy_id	Int	Stores enemy id for an adventure. Created as per the request from LevelGame class classes Enemy instance. Increase at each instance.
----------	-----	--

Table 2- Data Dictionary

23g Persistent Data management

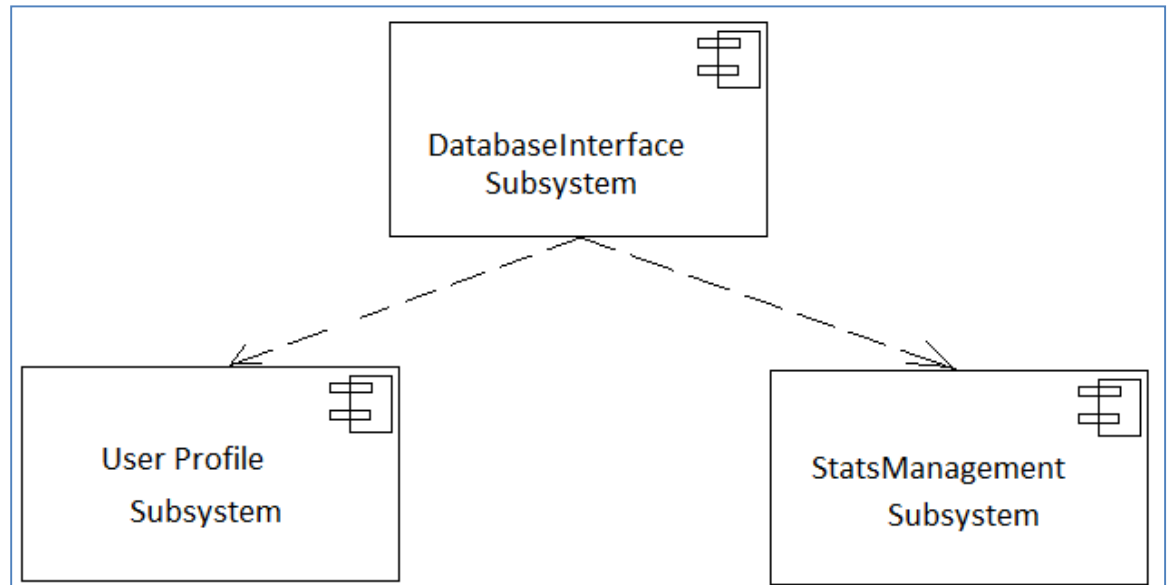


Figure 15 – Persistent Data management Diagram

User Profile Subsystem: The User Profile Subsystem is responsible for storing account information such as the end users email and password. Because this subsystem is used for accessing the account it supports fast storage and also uses multiple accesses. A relational database model is used for concurrency control.

StatsManagement Subsystem: The StatsManagement Subsystem is responsible for storing past game statistics for the Use Profile Subsystem. Some of the persistent data to be stored is how many games have been played, the maximum score, and the longest game duration. This subsystem supports statistics for all past games played. Because this subsystem supports multiple characters it will support multiple concurrent character statistics. A relational database model is used for concurrency control.

23h Access control and security

A	Authenticate
B	Add User

R	RequestStats
I	Interact
D	DisplayStats
G	GamePlayLogic
U	UpdateStats
RS	ReturnStats
X	Allow/Deny Updates from Database

<u>Objects ></u> Actors v	User Interface	User Profile	GamePlayManagement	Database Interface	Web Collector
Enterprise Administrator	A I	A B	R D G	U RS	R RS X
Client	A I	A B	R D G	U RS	R RS

Table 3- Access control Matrix

23i Global software control

This system uses ‘Event Driven’ global software control. The system waits for the user input/actions and it then reacts based upon the user request.

To ensure a robust system design, we propose the following strategic goals:

Database Connections: Be sure to open database connection on demand, and close the connection once you have acquired the desired task.

Data validation: Prior to query to database, be sure to sanitize the user input (if required) and use prepared statement to avoid system crashes due to unexpected user input.

23j Boundary conditions

Start-up System:

New Users: Upon accessing the adequate URL for our product, new users will be required to create a new profile. This is necessary because the system may be used by multiple individuals on the same workstation, each having their own statistics and character details. Once they have established their own account, then they will be provided with a character with default statistics i.e. XP = 10, Intelligence = 100 and Strength = 100. Also, user can form group and start playing the adventure.

Existing User: Upon accessing the adequate URL for our product, existing users will be asked to authenticate. Upon successful authentication, the system will retrieve the user details from the database and this will be reflected in the user interface. Also, user can form group and start playing the adventure.

Exit System:

User will use the exit functionality of the main interface at any time. In the event that there s unsaved user statistics/game play, the user will be presented with a dialog prompt with an option to save the changes before exiting. Additionally, if there s an ongoing database update at the time the exit button is pressed, the user will be presented with an option to wait for the update to complete.

Exception Handling:

The product shall handle exceptions in a way such that the system meets the design goals mentioned in section 21a. Exceptions like: Attempt to Signup using an existing email id, trying to exit while a database update operation is in progress, and many more.

General System Maintenance:

The product must not face a server down time of more than 15 minutes. Additionally, the product updates and required patches must be installed during a window of 3 am (GMT) to 4 am (GMT) and should not take more than 5 minutes as the online user impacted are least during this window.

24 Subsystem services

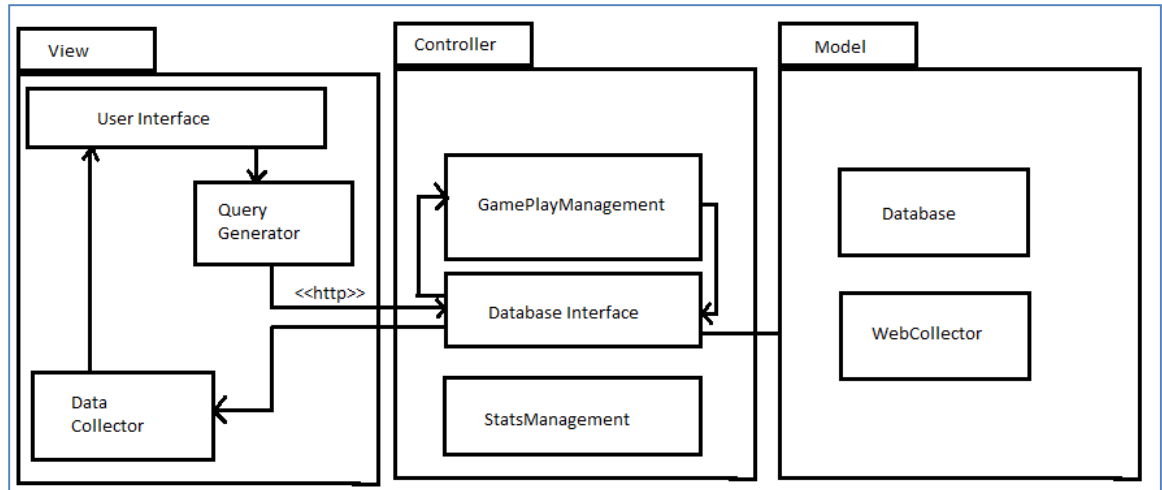


Figure 16 – Subsystem Services Diagram

25 User Interface

Content

The user interface will include the following menus: Login, Lobby, Group Formation, Adventure and Post Adventure Statistics.

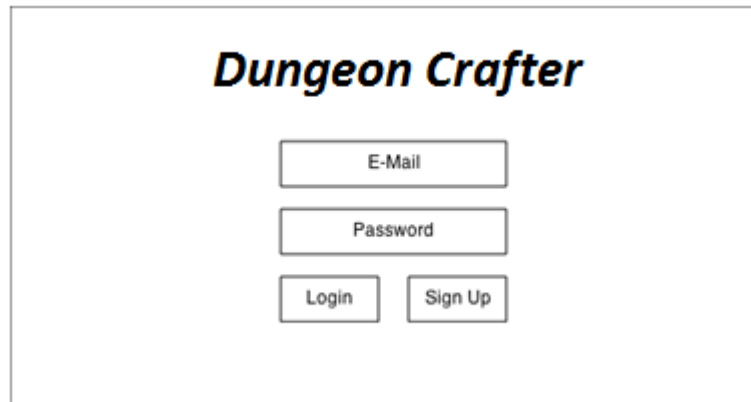
This interface will be designed around the theme of the game. It will include dark, medieval themes to contribute to the user's immersion. The background will be black, with stone textured borders that are reminiscent of an old dungeon. The font will be similar to those seen in old medieval texts. The buttons that will navigate the user to any adjacent menus will be lined up at the top of the screen, right under the "Dungeon Crafter" logo.

NOTE- The layouts shown for each User Interface section are very basic and simple keeping in mind the difficulty that will be faced by developers (students in next semester). As per the discretion of development team a better and enhanced set of graphics will be appreciated.

Login:

- This is where the user enters their email and password; these two fields will be in a box with two buttons: "Login" and "Sign Up". Below the login button will be a smaller link for users who have forgotten their password.
- Users who select "Sign Up" will be prompted to enter their email address, as well as set a password. The application will check if the email entered is already in the database, if it is not, a confirmation email will be sent to the user. After the user follows the link in this confirmation email, they will be allowed to log in with those credentials.

- “Dungeon Crafter” will read in large medieval font across the top of the screen, with game art in the background.
- An appropriate error message will be displayed if incorrect credentials were entered.
- If credentials are valid, the “Login” button takes the user to the next menu: Lobby.

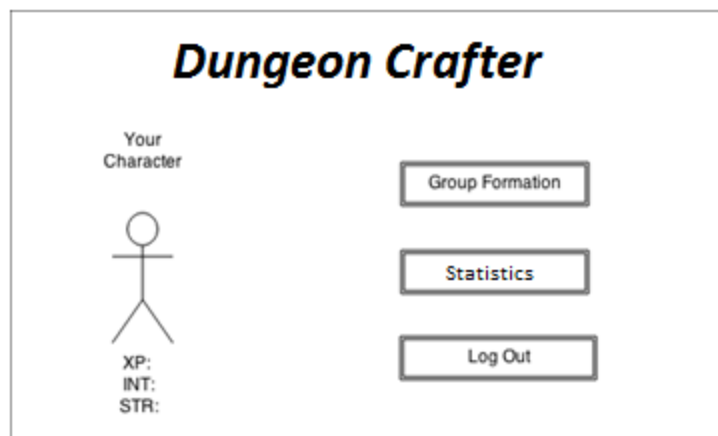


The image shows a user interface for the 'Dungeon Crafter' game. At the top, the title 'Dungeon Crafter' is displayed in a large, bold, medieval-style font. Below the title, there are three input fields: 'E-Mail', 'Password', and a 'Login' button. To the right of the 'Login' button is a 'Sign Up' button. The entire interface is enclosed in a rectangular border.

Figure 17 – Login/Signup User Interface Design

Lobby:

- Here the user will be able to view their character's XP, Intelligence, and Strength, along with all the equipment their character has.
- All character attributes in the Lobby are read only and cannot be modified in this menu.
- The available buttons at the top for the user to press include: Logout, and Group Formation.



The image shows a user interface for the 'Dungeon Crafter' game's lobby. At the top, the title 'Dungeon Crafter' is displayed in a large, bold, medieval-style font. Below the title, on the left side, there is a section titled 'Your Character' with a stick figure icon. Below the icon, the character's attributes are listed: 'XP:', 'INT:', and 'STR:'. On the right side, there are three buttons: 'Group Formation', 'Statistics', and 'Log Out'. The entire interface is enclosed in a rectangular border.

Figure 18 – Lobby User Interface Design

Group Formation:

- Users will be randomly grouped with other people who are searching for a group at the same time.
- The menu will display a “Searching for group” message until a group has been found.



Figure 19 – Group Formation User Interface Design

Adventure:

- The adventure will be displayed on a square grid, with all appropriate graphics (Level of adventure (1-100), XP, intelligence, and strength of player.).
- A unique graphic will be assigned to all users to denote their location on the grid.
- A button at the top right will allow users to “Quit” from the current adventure. When a player clicks this, a confirmation message will appear, informing the user that all progress will be lost, giving them the options “Quit” or “Back”.
- If a user quits from the adventure, a message will be displayed to all the remaining users in the group that he/she has quit.

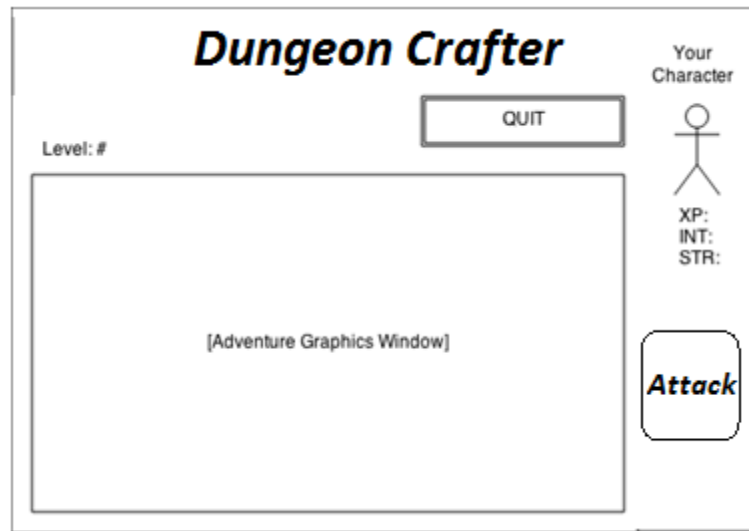


Figure 20 – Adventure User Interface Design

Post Adventure Statistics or Statistics window:

- This menu will display the following information from the adventure the user just completed:
 - o XP gained
 - o Intelligence gained
 - o Strength gained
- A button at the top will take the player back to the Lobby, or Exit the application once they have finished examining the statistics.

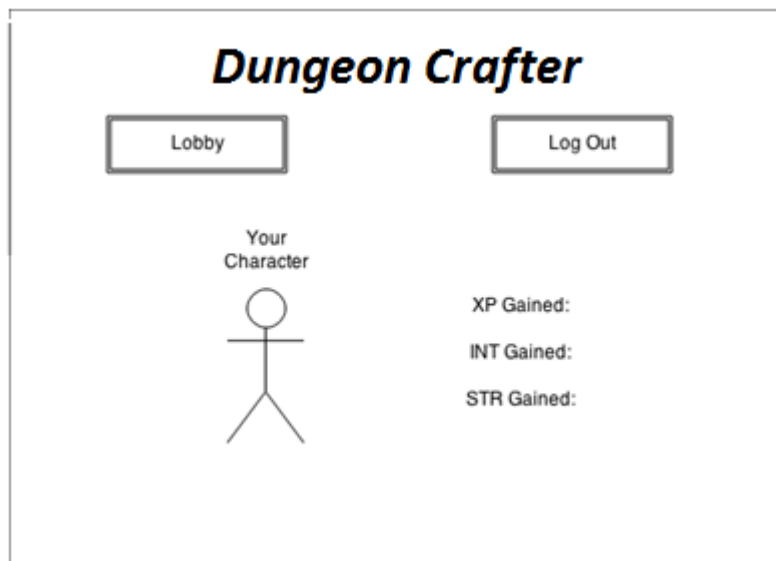


Figure 21 – Show Statistics User Interface Design

Motivation

The main goal of the user interface is to maximize immersion while being as easy to learn as possible. New users should be able to intuitively figure out how to navigate the menus in a relatively short amount of time.

Considerations

The user interface will be web based, and should run without problems on all of the major browsers (IE, Chrome, Safari, and Firefox).

Example

The designs of the old Run escape game, although this application will be much less graphics intensive.

26 Object Design

26a Object Design trade-offs

Content

The application will be free-to-play and web-based. This has proven to be a successful model for attracting new players.

Trade-offs -

- Free-to-play will have lower profit margins initially
- Web-based will limit the intensity of the graphics
- Advertisements may interrupt the user's immersion
- Game relies on internet connectivity of players

Motivation

Making the application web-based eliminates the need for users to install anything on their machines. This makes it more likely that a person will be willing to play the game. Also, being free to play allows anyone who visits the site to potentially start playing the game. This is especially important since a large portion of the project's success depends on the size of the community.

Considerations

To attract investors, there needs to be some sort of income for the maintainers of the game (if it is free-to-play). This could come in the form of advertisements in the game.

Since the game will be web-based, consideration of different web browsers needs to be taken. Ensuring that the program works on the most

common web browsers should be sufficient, and it may be worth advising the user to switch to a more common web browser if the client detects they are using a more obscure one.

26b Interface Documentation guidelines

Content

The most important aspects of the interface are:

- Immersive: Graphics and sound should set the mood for the game.
- Learnable: The interface should not intimidate new players.
- Efficient: The interface should be free of any unnecessary information.
- The naming convention and the visibility of each interface class attribute must be as per the [Naming convention table](#) and [visibility described in class diagram](#). More details are mentioned in next section.

Motivation

The game needs to attract new players while retaining existing players. Therefore, the interface should be easy to learn, as well as easy to remember. Navigation of the menus should feel natural and familiar to the user. Although the aesthetics are of very high importance to the design of the interface, it should not greatly affect the performance of the application.

Considerations

The audience of this game has a large age range. Therefore, the interface should be learnable to the standards of a young child, but not trivially dumbed down for the adult audience.

Example

A quick walkthrough of menu navigation and game play can be optionally viewed by a new player after they have registered their account.

26cPackages

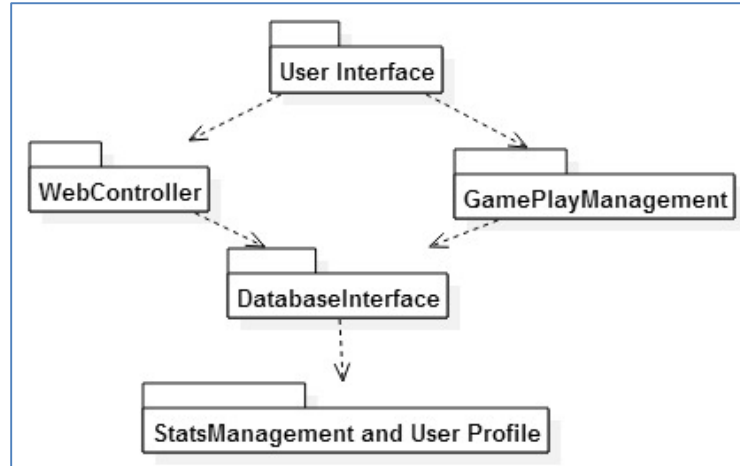


Figure 22 – Package Design

- User InterfacePackage
 - This package consists of the front-end of the application that every user interacts with. It consists of classes like – LoginForm and LobbyForm.
- GamePlayManagement
 - The back-end package that enforces the game rules, loads necessary adventure data, and determines entry, exit conditions. This package consists of following classes – GroupFormation, Adventure, LevelGame, Enemy, PlayerStats and DisplayStats.
- Web Controller
 - Handles all internet-based requests, such as authenticating the login, matching players together, and synchronizing the adventure among all the players. This package consists of following classes - PlayerAction
- DatabaseInterface
 - Handles all the data that needs to be stored for the game. Such as user information, character information, and adventure data files. This package consists of DB connector class.

Content

The user is presented with a graphical user interface by accessing the application through the web.

User accounts are stored in a User Info Database in the Database Tier. This information is checked when the user attempts to log into the application. A new row is added to the database when a user signs up.

A database holds the information of each user's 'character' that is tied to their account. This information gets modified (improved) every time the user completes an adventure. This information is updated in the Statistics Database at the moment of completion.

This important attributes and functions of each class are as below:

Note – Visibility is shown by: (+) for Public, (-) for Private , (#) for Protected

1. Player:

<u>Name</u>	<u>Type</u>	<u>Description</u>
Attributes		
+Name	String, Field Size = 15	Name of the user
+Email_id	String , Field Size = 15	Email id of the user
#Password	Password, Field Size = 10	Password of the user
+Strength (ST),	Int	This is one of the skill that player uses to fight with the enemy. Initialized to 100.
+Intelligence (IN)	Int	This is one of the skill that player uses to fight with the enemy. Initialized to 100.
+Experience (XP)	Int	Experience (XP) Initialized to 10

Functions		
+ValidateUserCredentials(email_id, password)	Int	Validates the user credentials.
+PlayGame()	Int	Act as the interface for the user to interact during the game play.

2. LoginForm:

<u>Name</u>	<u>Type</u>	<u>Description</u>
Attributes		
+Email_id	String , Field Size = 15	Email id entered by the user
#Password	Password, Field Size = 10	Password entered by the user
Functions		
+ValidateCredentials(email_id, password)	Int	Validates the user credentials.

3. LobbyForm:

<u>Name</u>	<u>Type</u>	<u>Description</u>
Functions		
+Exit	Void	Exits the current login.
+GroupFormation()	Void	Navigates user to Group formation screen.

4. DBConnector:

<u>Name</u>	<u>Type</u>	<u>Description</u>
Functions		
+ConnectUserLoginDB	Void	Creates connection to UserLogin DB where user's login details are stored.
+ConnectUserStatsDB	Void	Creates connection to UserStats DB where User's stats are stored.

5. DisplayStats:

<u>Name</u>	<u>Type</u>	<u>Description</u>
-------------	-------------	--------------------

Attributes		
#Strength (ST),	Int	This is one of the skill that player uses to fight with the enemy. Initialized to 100.
#Intelligence (IN)	Int	This is one of the skill that player uses to fight with the enemy. Initialized to 100.
#Experience (XP)	Int	Total Experience (XP) gained by a user. Initialized to 10
Functions		
+DisplayStats(Email_id)	Void	Displays stats for a user as per the user credential passed i.e. Email-id.

6. GroupFormation:

<u>Name</u>	<u>Type</u>	<u>Description</u>
Attributes		
+PlayerCount	Int	This attribute keeps count of the number of player that is connected.
Functions		
+PlayerCount()	Void	This function checks for the number of player currently willing to form a group
+InitiateAdventure()	Void	Instantiates adventure class if the player count reaches to 5. There by beginning a new adventure.
+NotifyCountToPlayer()	Void	This function displays the group count i.e. the number of people that are willing to join the current adventure.
+Timeout()	Void	This function check if the user has been

		waiting for more than 5 minutes, if yes, then prompts the user to choose if he wants to wait for group formation or exit.
--	--	---

7. PlayerStats:

<u>Name</u>	<u>Type</u>	<u>Description</u>
Attributes		
#Strength (ST),	Int	This is one of the skill that player uses to fight with the enemy. Initialized to 100.
#Intelligence (IN)	Int	This is one of the skill that player uses to fight with the enemy. Initialized to 100.
#Experience (XP)	Int	Total Experience (XP) gained by a user. Initialized to 10
Functions		
+UpdateStats(Email_id)	Void	Updates the UserStats DB with the new skills; user is mapped using email id.
+Fetchand ReturnStats(Email_id)	Int	This function fetches the requested stats and returns the same.

8. Adventure:

<u>Name</u>	<u>Type</u>	<u>Description</u>
Attributes		
#Adventure_id	Int	This attribute is used to identify an adventure uniquely.
Functions		
+LevelCountCheck()	Void	This function checks the level count, so that an adventure is complete when the

		count reaches 100.
+NewLevel()	Void	This function instantiates a new level class when the previous class is successfully completed.
+MAnagePlayerStats()	Void	This function manages the user stats i.e. if the user successfully cleared a level or eliminated an enemy, then the user stats are updated with the new increment as per the game rules mentioned in the Project Description section.
+UpdateStats()	Void	This function requests an update of the player stats if the adventure is completed by the user.

9. LevelGame:

<u>Name</u>	<u>Type</u>	<u>Description</u>
Attributes		
#Level_id	Int	This attribute is used to identify a level in an adventure uniquely. Initialized to 1.
+Level_Status	String, length = 10	This attribute stores the stats of the current level i.e. whether the level is complete or not.
Functions		
+CreateEnemy (int)	Void	This function instantiates an Enemy class if all the players have arrived to a new level.
+PlayerAction()	Void	This function instantiates a PlayerAction class which listens for an action from each

		player.
EvaluateAttack(Email_id)	Void	This function evaluates the attack made as per the PlayerAction, i.e. the skills of the player who attacked are evaluated against the skills of the enemy.
+LevelStatus()	Void	This function manages the status of the current level. If complete then updates the attribute Level_satus to 'complete'.

10. Enemy:

<u>Name</u>	<u>Type</u>	<u>Description</u>
Attributes		
#Strength (ST),	Int	This is one of the skills that Enemy uses to fight with the enemy. Value depends on the level_id.
#Intelligence (IN)	Int	This is one of the skills that Enemy uses to fight with the enemy. Value depends on the level_id.
#Enemy_id	Int	This attribute uniquely identifies the instance of the enemy created.
Functions		
+CreateEnemy(Level_id)	Void	This function creates an enemy as per the level id. The logic of enemy creation is as per the description in project description and requirements section.

Motivation

Storing character information in a database is safer than storing this information locally, as it prevents users from illegally modifying their character information. It also guarantees that their character will be saved.

Considerations

Since web standards change, the application may encounter bugs and crashes as new versions of web browsers are released. This may require a team of developers to make changes to the code of the application when these new versions are released.

Example

By examining the Google Chrome change log, one of the internet's most popular browsers, it is obviously apparent that changes are made to web browsers on a very regular basis.

IV Test Plans

27 Features to be tested / Not to be tested

Content

Features to be tested (in addition to the [Requirements](#))

- The products ability to interface with any off the shelf solutions.
- The product's ability to maintain a stable network connection.
- The product's ability to allow sign-up and maintain user's information on databases for login verification and user progress
- The product's ability to execute an adventure file with visual representation
- The product's ability to acknowledge exit conditions of adventure
- The product's ability to display accurate statistics when requested
- The products ability to meet [performance requirements](#).
- The products ability to meet [Dependability Requirements](#)
- The products ability to meet [Maintainability and Supportability Requirements](#).
- The products ability to meet [Security Requirements](#).
- The products ability to meet [Usability and Humanity Requirements](#).
- The products ability to meet [Look and Feel Requirements](#).
- The products ability to meet [Operational and Environmental Requirements](#).
- The products ability to meet [Cultural and Political Requirements](#).
- The products ability to meet [Legal Requirements](#).

Features Not To Be Tested:

- The product's ability to restrict a user from multiple accounts
- The product's ability to store adventure files as data files
- The product's ability to contact support for user login support.

Motivation

From the user's point of view, we need to ensure that the user is able to enjoy the full worth of the product without having to worry about the functionalities and the implementations behind the output.

Example

If user loses internet connection and is wondering why the product is not working, then the troubleshoot web page would suggest the user to check internet connection.

28 Pass/Fail Criteria

Content

Tests executed on components only pass when they satisfy the signatures, constraints, and interfaces dictated by the Object Design Specification for that component. This includes positive tests, negative and stress tests, and boundary tests.

If a test exhibits a product failure to meet the objectives of the object design specification, it will fail and a defect/issue will be reported.

Motivation

We need to make sure that all predetermined test cases are met in order that the product may work to its full capacity and serves its intended purpose.

Considerations

Some tests may work under some conditions and not work under some other conditions, hence all tests are done multiple times under various conditions to confirm that part of the application really works fine.

Example

A user may be able to retrieve information from past login while device is powered, but not on occasion of a power down, all information may be lost.

29 Approach

Content

General Testing Strategy

Unit testing and component testing will be performed on the components as they are developed. Test will be executed using test code in the form of either custom test tools or as an automated suite of tests run against the components in their individual sandboxes.

The build acceptance test and the unit test suite will be used as a regression during the integration of components. However, as the integration begins to include GUI level functionality, the tests being run will require more manual testing and less automated testing.

Motivation

The goal is to give the user maximum satisfaction, hence we test all cases that may pertain to a user of the product. Problem logs are used to keep each problem and its full details documented for future repairs.

Considerations

Running test should not take so much time in order to get other final phases of the product development done, hence all testing should have a time frame they must adhere to.

Example

A situation whereby so much time is spent on one test and there is a problem with some other component that has not been tested. The problem will be found late and will delay the eventual release of the product.

30 Suspension and resumption

Content

Unit Test Suite:

Tests components as they are developed, Tests will focus interfaces of the components and low-level tests will be executed on the remaining underlying component classes.

Suspension: When Unit Test reports a failure during component testing throughout development testing will suspend.

Resumption: Resumption occurs when current build is analyzed and reported problematic components are resolved/repaired.

Build Acceptance Test: When a build is considered ready for testing, a broad but shallow set of tests will determine the stability of the current build (Occurs after Unit Testing)

Suspension: If the build test fails. All further testing will be suspended.

Resumption: A new build with appropriate repairs is prepared and ready for a reevaluation by Build Acceptance Test suite.

Regression Test: Build by build basis test that reviews major bug fixes and code changes to determine any effect/risk it may have on the system.

Suspension: Testing will suspend if Regressions determines a high effect/risk on the system.

Resumption: Once all risks/effects are dealt accordingly a reevaluation of the Regression Test is required. A “pass” is required by the second Regression test for all further testing to resume.

System Design Changes: If throughout development of the project issues arise that are better repaired by altering the design of the system architecture, object design, requirements

Motivation

Time is a very treasured asset in building this product, and in order to utilize it efficiently, we need to be able to suspend all unproductive tests. Also, the whole test team must be up to date on which tests are being executed and which have been suspended, so there should be a suspension of testing while there is a public holiday until everyone is back at work.

Considerations

We want all the test team members up to date on which tests have been suspended and which have resumed, hence there should be a general notification sent to all test team members on occasion of a suspension or resumption.

Example

A situation where by a test team member is working on a suspended test could lead to long hours wasted which could have been dedicated to something more productive.

31 Testing materials (hardware / software requirements)

Content

Hardware Requirements: Four copies of the system under test. The hardware components in the system, in a single set are a Database Server, a Web Server, a client PC with a supported web browser, and the embedded system that is used. four systems allow the team to test several components in parallel.

Software Requirements: The database server needs the appropriate database installed, setup, and configured properly. The client machine needs a supported browser installed and properly configured

Motivation

All tests need to be executed under the same condition in order to minimize differing results and to narrow down the problem on occasion of differing results.

Considerations

The test team members will be working together most of the time, hence they do not require individual offices. One office that can accommodate all the test team members is most adequate for their task.

Example

A situation where test team member want to compare results or analyze things together, they need a conducive room that can accommodate them and enable them to work together effectively,

32 Test cases

To demonstrate the granularity of the test case that needs to be developed, below are few of the important test cases:-

Test Cases for nonfunctional requirements (Nftc)

Test case specification identifier: 1

Test Case Name: Nftc_usability_GuiNavigation

Input specifications: User clicks on menu changing button

Output specifications: Appropriate menu appears without distortion or alterations.

Test case specification identifier: 2

Test Case Name: Nftc_usability_EaseOfUse

Input specifications: User can navigate through User Interface and correctly perform all Game Mechanics without much effort.

Output specifications: Appropriate menu actions execute and user content with control scheme.

Test case specification identifier: 3

Test Case Name: Nftc_usability_UserManual

Input specifications: User clicks on the user manual, and navigate through the user manual.

Output specifications: User manual opens up. User manual is detail and provides detail information with text and screen shot about the system.

Test case specification identifier: 4

Test Case Name: Nftc_reliability_ErrorRecovery

Input specifications: Perform all possible actions accepted in the application. Perform action in the application that is not supported by it.

Output specifications: The application is reliable in terms that actions supported by user are performed without crashing the application, and actions not so supported by the application displays a user friendly error message.

Test case specification identifier: 5

Test Case Name: Nftc_reliability_CustSupp

Input specifications: User inquiring about the use of product, and other concerns associated with our product.

Output specifications: Provide quick and accurate responses and perform any other relating task to ensure customer satisfaction.

Test Cases for functional requirements (frtc)

Test case specification identifier: 6

Test Case Name: Frtc_login

Input specifications: The system requests that the user enter his/her username and password. The user enters his/her username and password.

Output specifications: The system validates the entered username and password and logs the user into the system.

Test case specification identifier: 7

Test Case Name: Frtc_loginAlt

Input specifications: The system requests that the user enter his/her username and password. The user enters an invalid username and password.

Output specifications: The system invalidates the entered username and password and displays an error, and returns to the basic flow defined in the use case.

Test case specification identifier: 8

Test Case Name: Frtc_SignUp

Input specifications: The system requests that the user enter his/her email, first and last name, age, class, username and password.

Output specifications: The system creates the entered username (if available) stores password and logs the rest of the collected information to our database system.

Test case specification identifier: 9

Test Case Name: Frtc_GamePlay

Input specifications: User interacts with GUI and clicks on a clickable item which sends a request to the system server for correct response.

Output specifications: system server acknowledges and attempts to execute request. Error message returns if request cannot be performed. Depending on request GUI might be updated or stats may be changed. If user defeats enemy then his/her stats should get updated, also energy should be '-1'. If user loses then he/she should be eliminated from current adventure.

Test case specification identifier: 10

Test Case Name: Frtc_ExitFuncs

Input specifications: An exit has been requested by the user /game play in a certain menu page

Output specifications: acknowledge and execute corresponding exit condition as requested by user/gameplay. Follow exit protocols if request fails to output correctly.

Test case specification identifier: 11

Test Case Name: Frtc_display_gameplay

Input specifications: sends request for visual representation of user graphical menu input to servers

Output specifications: correct visual representation is displayed without distortion or alteration.

Test Cases for Network/Database Operations (dbtc)

Test case specification identifier: 12

Test Case Name: Dbtc_updateUser

Input specifications: user stats/information has changed and a request has been sent to sync new data to the database

Output specifications: appropriate message regarding pass/fail of user stats/information attempt to save successfully against database.

Test case specification identifier: 13

Test Case Name: Dbtc_autheticateLogin

Input specifications: a user has attempted to log in with a specific username and password, request is sent to database

Output specifications: validates credentials sent in with request against most up-to-date user login database. Sends corresponding message upon pass/fail.

Test case specification identifier: 14

Test Case Name: Dbtc_requestStats

Input specifications: a request for the user's most up-to-date statistics has been made by game play

Output specifications: sends correct statistic file and visual representation of the stat file.

Test case specification identifier: 15

Test Case Name: Dbtc_saveStats

Input specifications: new Stats have been detected and need to be updated with server database

Output specifications: new stats are updated successfully.

Test case specification identifier: 16

Test Case Name: Dbtc_display_gameplay

Input specifications: frtc_display_game request

Output specifications: translates frtc_display_game requests with the visual representation of request.

Test case specification identifier: 17

Test Case Name: Dbtc_createUser

Input specifications: stores new user sign up information in database

Output specifications: successfully allows credentials to be verifiable for future log-ins.

33 Testing schedule

<u>Task Name</u>	<u>Start</u>	<u>Finish</u>
Test Planning	Beginning of month	Same day
System testing	Week 1	End of week
Regression testing	Week 2	End of week
Resolution of final defects	Week 3	End of week
Performance testing	Week 4	End of week

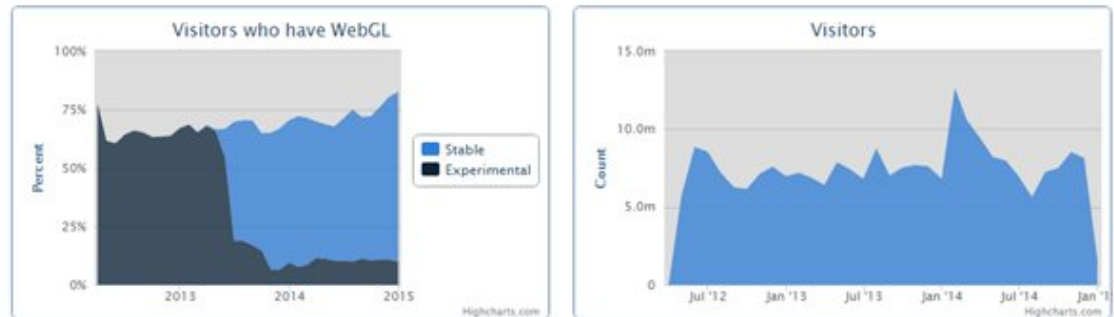


Figure 24 – Statistics of visitors using WebGL

Adobe Flash

Flash is a multimedia software platform for rich internet applications. Flash is compatible with Windows, OS X, and Linux platforms as a free package. However, all users must have the Flash plug-in installed in order to use a Flash application. However, it has recently been noted that there are many security risks associated with Flash applications

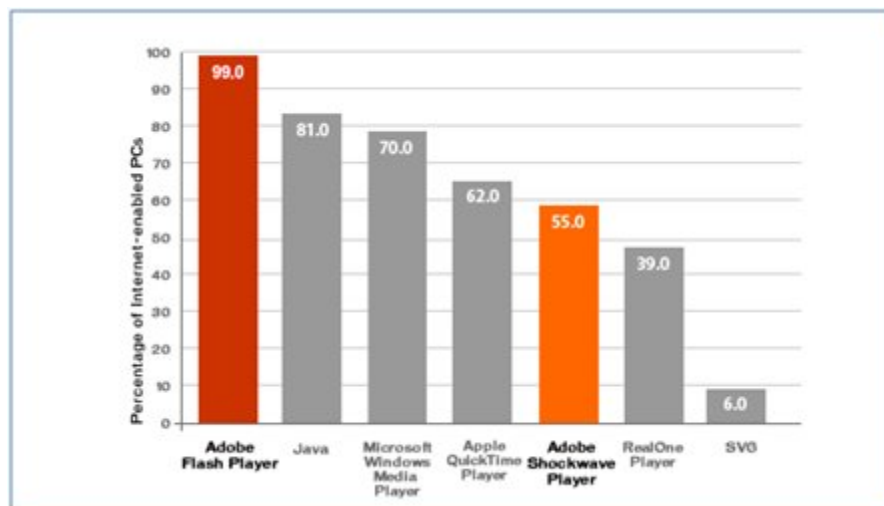


Figure 25 – Percentage of Internet-enabled PCs having various player

35b Reusable Components

No reusable components available.

35c Products That Can Be Copied

No products available that can be copied.

36 New Problems

36a Effects on the Current Environment

Below is a list of potential problems the user may face while using Dungeon Crafter:

- Timing out while searching for a group
 - If the user happens to be the only online user searching for a group for more than 1+ minute, the user will be timed-out.
- Disconnecting during a game
 - If the user loses communication with the server while he/she is actively in a game, the user will lose everything he/she acquired during that session.
- Forgetting their password
 - The user may forget their log-on password to access their account.

36b Effects on the Installed Systems

No Impact on installed Systems.

Reason – It's a web based game.

36c Potential User Problems

No Impact on existing users.

Reason – It's a web based game.

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

Anticipated Limitations:

- The server will only be able to handle a finite number of online users playing the game.
- The adventure database will have a limited capacity of adventures.
- The game's web graphics will be limited by the browser's allowed GPU acceleration.

36e Follow-Up Problems

- How much leeway should there be between the actual server capacity and the expected number of online users.
- When should the server's capacity be expanded?

When should the memory of the adventure and user database be increased?

37 Tasks

37a Project Planning

Dungeon Crafter will follow the following Agile Life Cycle for development:

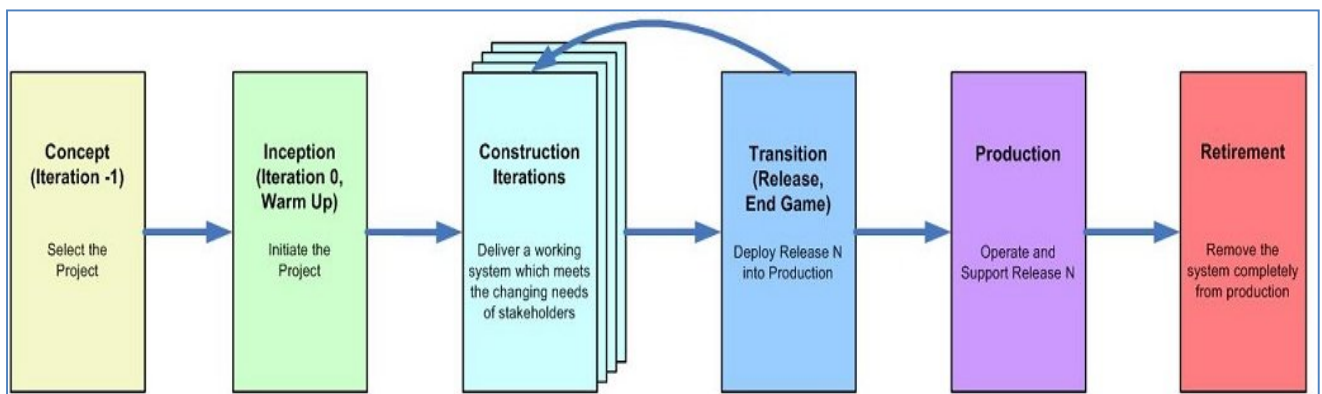


Figure 26 – Agile Life Cycle Model

37b Planning of the Development Phases

Concept

- Identify potential projects
- Prioritize potential projects
- Develop initial vision

Inception

- Active stakeholder participation
- Obtain funding and support
- Start building the team
- Initial requirements envisioning
- Initial architecture envisioning
- Setup environment

Construction Iterations

- Active stakeholder participation
- Collaborative development

- Model Storming
- Test driven design
- Confirmatory testing
- Evolve documentation
- Internally deploy software

Transition

- Active stakeholder participation
- Final system testing
- Final acceptance testing
- Finalize documentation
- Pilot test the release
- Train end users
- Train production staff
- Deploy system into production

Production

- Operate the system
- Support the system
- Identify defects and enhancements

Retirement

- Remove the final version of the system-data conversion
- Migrate users
- Update enterprise models

38 Migration to the New Product

38a Requirements for Migration to the New Product

Migrating/Installation of the product onto a suitable system requires a functional keyboard & mouse, and stable Internet connection and a compatible Internet Browser.

The browser will typically handle rendering the graphical interface once a user logs onto the site. Sufficient disk drive space is required

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

Graphical Interface requests will need to translate to the visual representation that will be displayed to the end-user.

User Log In information need to be verified and the Internet browser may keep cookie information for ease of use of the project to interpret user details such as progress and skill set.

39 Risks

(30% Likely) Time conflicts –Project member's inability to allot time towards project to meet deadlines, and make progress. Time impact variable.

(30% Likely) Not Meeting/ Extending Deadlines – Discovering portions of the project that are either more time consuming than originally thought and having to devote more time. Time impact variable.

(25% Likely) Budgeting errors – over spending, insufficient funds to cover monthly maintenance costs. Cost impact variable.

(20% Likely) Management malpractice – Miscommunication and disputes can cause delays in product development. Time impact variable.

(5% Likely) Third Party services utilized in system become unavailable during testing – Inability to use third party services can hinder testing/development. Time impact variable

40 Costs

Estimated cost of project:

Developer(s) salary - \$150000

Recurring server maintenance (including staffing and possible third party management services) -70,000/year

Legal fees, licensing, copyrights – 25000/year

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

-Offline multi-platform desktop version of the project

-Improved game play mechanics (New enemies, dungeons, items, difficulty settings), graphics, lower resource costs.

-Multiplatform mobile versions of the project

-More in-depth character stats, customization.

Motivation

To allow room for expanding possible franchise of this project it may improve and expand customer base. Keep users satisfied with produce.

42 Ideas for Solutions

Content

- Maintaining important components of game play and overall server/network functionality of the software design as web-based components. Ensure resource consumption is efficient among all tasks.
- Allow visual representation to scale accordingly or maintain constant frame size.
- Allow universal server access between different platforms of the project.

Motivation

- To allow porting of the game to different platforms to be easier.
- To allow efficient use of resources and to maintain lowest possible budget.
- To allow more efficient data collection and storage

43 Project Retrospective

- Rigorous testing throughout the development process to ensure the least amount of bugs will allow for stable releases. Testing thoroughly will allow for better maintenance of overall project and potentially better insight when new bugs appear, as developer(s) would have a more extensive understanding of the project.
- Allowing enough time between sprints and releases to ease any potential deadline stress while making notable progress overall

VI References / Bibliography

- [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.