Snapster

CSE3444 – Software Requirements Project

Spring 2022 – Section 001 - University of North Texas

Manushi Parajuli, **Muhammad Ghazi, Steven Jamaica**

1. Background

Software Overview

The SNAPSTER app is being developed to provide a decentralized social media platform for sharing videos and audio content. The purpose of the software is to allow users to easily share and browse content from their mobile devices, with no centralized data storage.

The app will include additional features such as user accounts, friending, avatars, chat/DM ability, video/camera recording, and custom filters. Users will be able to earn points by sharing content, which can be used for avatar customization or purchased directly from the company. The app will run on mobile platforms such as iOS and Android. The application is relatively novel in its decentralized approach to social media, and there may be technical risks associated with the point-to-point file sharing and parallel download features. External inputs and outputs will include user-generated content, updates to custom filters from the company repository, and advertisements from sponsors.

The quality attributes of the app will need to be considered, such as reliability and security of user-generated content. The development team will include architects, engineers, and other technical professionals responsible for system development and operation. The team will utilize various development, integration, and testing tools to ensure the app is functional and reliable.

Organization / Environment Overview

Muhammed Ghazi is the project manager and will be overseeing the project to ensure its completion on time and within budget. The project manager will be the bridge between stakeholders, investors, and customers. He is also responsible for working closely with the team to ensure efficient development.

Steven Jamaica is the lead developer and will be responsible for writing efficient code with minimal bugs. The team specializes in both front-end and back-end development. The lead developer works closely with the developer, project manager, and tester to ensure the specifications and requirements are met. The developer is under the development team.

Manushi Parajuli is a QA tester and is responsible for testing the quality and functionality of the app to meet the project requirement. The QA is responsible for creating test plans, executing, and reporting bugs/issues to the developer. They work closely with the developer to ensure the delivery of a high-quality application. The QA tester is also under the development team.

Snapster is headquartered in Texas to reduce the overhead cost of living. Around forty percent of the employees work remotely, twenty percent hybrid, and forty percent work on-site. Around twenty percent of the remote workers are outsourced overseas to reduce the time and cost of software development. As a startup company, Snapster hires mostly young developers with limited work experience. A few developers who are highly experienced (around 7-10) will be leading the team. The company will hire a few Computer Science interns to help the developers and testers. Most of the lead developers will be working on-site. The interns and inexperienced developers work hybrid. The overseas team and limited experienced developers will work remotely. The team will work five days a week. The overseas developers will be working during the US nighttime.

The development team (software developers, designers, and testers) is responsible for building and testing the software. The team of system administrators, network engineers, and database administrators will take care of operating the system. The development and operation team will also maintain the system in addition to building, testing, and operating it. The main groups of people involved in the system development are architects, engineers, designers, and QA testers. Architects are responsible for designing the overall system architecture including the hardware and software components. Software engineers are responsible for the coding part whereas UI/UX designer is responsible for designing the front end. The designer and the engineer on our project are in the same development team. QA testers are primarily responsible for detecting bugs and reporting to the developers.

In a high-level sense, the architects would communicate the system design to the developers and the developers will write code accordingly. The designers and developers collaborate to create an attractive and convenient user interface. The QA testers provide feedback to the development team to improve the app's quality and identify and report issues/bugs. All these groups work together to ensure the timely delivery of the app while meeting user expectations.

Schedule / Effort Overview

For the development of Snapster, project plans like Work Breakdown Structure (WBS) and Basis of estimation (BOE) are crucial. The project schedule to achieve milestones are planning, design, development, testing, and deployment.

In the beginning, around thirty main developers and designers might be enough to develop the Snapster. Around 5 testers are necessary. Programming languages and frameworks like Java, React, Swift, etc. will be used. IDE like Android Studio and Eclipse will be used. MySQL and MongoDB to store, manage, and retrieve the data. Cloud hosting services like AWS will be utilized.

Breaking down of the project into smaller and manageable tasks will simplify the project and increase the overall efficiency. So, we will assign the task to a specific team lead who will ensure the project progresses smoothly. IP Multimedia Subsystem will be used to create a project schedule.

Management Overview

Technical and Managerial aspects of the program are as follows

Technically, the development of Snapster will require the utilization of various tools and technologies for various stages of the software development lifecycle. These tools may include programming languages like Java, Python, or Swift, as well as development environments such as Eclipse or Xcode. Also, various software testing and integration tools will be used to ensure the system's quality and performance, such as JUnit or Selenium for automated testing, and Jenkins or Travis CI for continuous integration. The technical strategy must be defined to ensure that the system is developed in a scalable, efficient, and secure way, with appropriate use of caching, load balancing, and security protocols.

Managerially, the project will require the coordination of various teams and stakeholders, such as developers, architects, project managers, and business analysts. An Agile development methodology may be used, with regular sprint planning, review, and retrospectives. Effective communication, collaboration, and decision-making will be essential for the project's successful delivery. Technical development risks may include issues related to data privacy, security, compliance, scalability, and performance. These risks must be identified, assessed, and mitigated throughout the project lifecycle.

2. Program Plan

Software Process Model

The Appropriate software process model to choose for the Snapster™ app would be the Unified Process Model. The app relies on user interaction as there is no centralized photo library or video gallery that is saved in the Snapster™ database. Since the users must generate and share content that would allow them to interact with their social circle and even meet others outside that social circle, then the entire structure of the app is built around their desire to express themselves and to communicate with others. The company essentially just has to provide updates to keep the application interesting and to accommodate for new fads that the users would like to see and interact with so that the company can retain users.

Communication: Initial communication for the inception of Snapster™ would be with the Stakeholders and Software Engineers. As the Project progresses and a working application is built it can be submitted into the two main markets for mobile apps, such as Google Play® and Apple’s App Store®. Once the application has had a run with the open public we prioritize the feedback and opinions of customers just as much as major stakeholders as the evolution and upkeep of the app will no longer rely just on stakeholder money.

Planning: The initial planning of the app begins with the software developers and the Stakeholders, as time progresses with a working application, the opinions of the users are taken into account and any final decision will have the feedback of the stakeholders, the developers, and the users.

Modeling: During initial modeling, prototypes can start being made and a general idea for the app's architecture can begin to take shape. Prototypes can be shown to stakeholders for opinions on how to move the project forward and any beginning bugs can be addressed. As time progresses any future models will be built upon past models and any features that weren't implemented well or disliked can be addressed and new prototypes can be formed.

Construction: The main features for the app's first deployment are made with the stakeholder's ideas in mind. These are prototypes that will have functions that aren't completely done being worked in and bugs will be present, they are addressed as they come and the main focus is to have features that can expand the base use of the app which is image and video sharing. Over time new features will be implemented in this stage and a test group will have an unreleased version of the app with new features and see how it begins to behave on a daily basis.

Deployment: At this stage there is a small group of people that will have an unofficial released project, the feedback from this group will help shape the official release to the public and any issues that weren't found in the modeling or construction phase can be addressed and the group can give a nice outside perspective of the use of the app. They are essentially a sample size of the overall public.

Software increment/Release: At this stage a working product has been made with minimal bugs and issues, the past prototypes and test cases have proven that the product is safe, usable, and appealing to the overall target audience, and any future changes will be made with the feedback of the day-to-day users and the wants of the stakeholders.

Team Decomposition

Novice: Steven Jamaica Cervera

Salary Range: $70,000 – $80,000

Would be responsible for the less intricate features such as; UI color, typeface, help options, the privacy feature that lets users decide what can be downloaded, the point system for users to share more content, the account settings for logging in, friend requests, chat and direct messaging options, and the content search menu.

Journeymen: Manushi Parajuli

Salary Range: $110,000 – $120,000

Would be responsible for more intricate features in the software such as; recording and local storage for media, implementing the filters from the company repository, the sharing of media to other friends including stories, downloading content into cache, the point of sale for the point base system, and implementation of advertisers in story review mode. They would also be responsible for the copyright notice notification.

Expert: Muhammad Ghazi

Salary Range: $158,00 – $168,000

Would be responsible for the oversight of the project and one of the main leaders in determining what features can be developed as well as the architecture of the software. They would be responsible for the decentralization of media being shared among users, this means they are responsible for the point-to-point system (seeding) that makes the media quicker to download with more people having that media saved on their devices, this saves the company from having huge database systems to store the unquantifiable number of media being shared. They would also be responsible for choosing the advertisers that would be implemented into the story review mode. Since they are involved with very important information, they are also responsible for the prevention of user information from being easily attacked and accessible.

Work Breakdown Structure (WBS)

Work Breakdown Structure Table
Provide basic information about the project including: Project Title – The proper name used to identify this project; Project Manager - the person with responsibility for the successful delivery of the project to time cost and quality. Prepared by – The person(s) preparing this document; Date/Control Number – The date the plan is finalized and the change or configuration item control number assigned.

Project Title:	**Snapster™**		**Prepared by:**	**Steven Jamaica Cervera**
Project Manager:	**Muhammad Ghazi**		**Date / Control Number:**	**2/28/2023**

Element Number	**WBS Elements Activity, Task, or Sub-Task Name**	**Definition of Activity or Task (Description)**	**Responsible Person or Group**	**Estimated (E) or Actual (A) Cost (Cross reference to budget)**
The unique reference ID for the activity or task.	*Enter the name or title of the Task, Sub-task, Activity or Deliverable.*	*Provide a brief description of this Activity, Task or Sub-Task.*	*Enter the person or group(s) who are responsible*	*Enter the estimate or actual cost of the activity or task or add a cross reference to the budget.*
1	**All Software**	Develop All Software	Software Team	500,000
1.1	**User Interface**	Complete User Interface	User Interface Team	75,000
1.1.1	Requirements Analysis	Easy UI for user interaction	Steven Jamaica	
1.1.1.1	Task 1	Must be easy navigate		
1.1.1.2	Task 2	Typeface is readable		
1.1.1.3	Task 3	Easy Profile Settings		
1.1.1.4	Task 4	Social options		

1.1.1.5	Task 5	Help Options		
1.1.2	Design	Design User Interface		
1.1.2.1	Task 1	Picture and chat options when opening app		
1.1.2.2	Task 2	No cursive fonts, unless it's a media edit option, or **overbolded**		
1.1.2.3	Task 3	Easy to change privacy, and avatar settings		
1.1.2.4	Task 4	Chat, Friend requests, and search options easy to find		
1.1.2.5	Task 5	Easy to contact support, or report issues		
1.1.3	Code & Unit Test	Implement code & Perform Unit Test		
1.1.3.1	Task 1	Picture and chat options respond to user touch and open with minimal delay		
1.1.3.2	Task 2	Fonts aren't overwhelming and options are easy to read and find		
1.1.3.3	Task 3	User can change the privacy settings for who can view their stories, and can customize their avatar to express their image		
1.1.3.4	Task 4	Friends are added by a special code and can be contacted to share images and videos. People and media categories can be		

		searched and found through key words.		
1.1.3.5	Task 5	The help option is in the user settings and will have contact information for the support team.		
1.2	**Data Management**	Complete Data Management of media	Data Management Team	115,000
1.2.1	Requirements Analysis	Data is handled in ways that make the app interactive and safe	Manushi Parajuli	
1.2.1.1	Task 1	Recording and local storage for media		
1.2.1.2	Task 2	Implementing filters from company repository		
1.2.1.3	Task 3	Media shared through users		
1.2.1.4	Task 4	Payment Acceptance		
1.2.1.5	Task 5	Advertisements		
1.2.1.6	Task 6	Copyright notice	Legal and Data team	5,000
1.2.2	Design	Design Data Management		
1.2.2.1	Task 1	Option to save taken media through app and able to upload media from local directory to app		

1.2.2.2	Task 2	Filters are pulled from company repository monthly for new user interactions		
1.2.2.3	Task 3	Media viewed by user is saved in temporary cache.		
1.2.2.4	Task 4	Money transaction for points		
1.2.2.5	Task 5	Implement pre-selected ads in user stories		
1.2.2.6	Task 6	There should be a copyright notice every so often, so that users know to check their local laws		
1.2.3	Code & Unit Test	Implement code & Perform Unit Test		
1.2.3.1	Task 1	Media can be uploaded from local storage and can also be recorded and saved to local storage.		
1.2.3.2	Task 2	Filters are available when selected		
1.2.3.3	Task 3	Media is sent to cache and can be cleared and written over automatically		
1.2.3.4	Task 4	Payment methods are accepted, and user can keep the points bought for them		
1.2.3.5	Task 5	Different Ads chosen from upper management are		

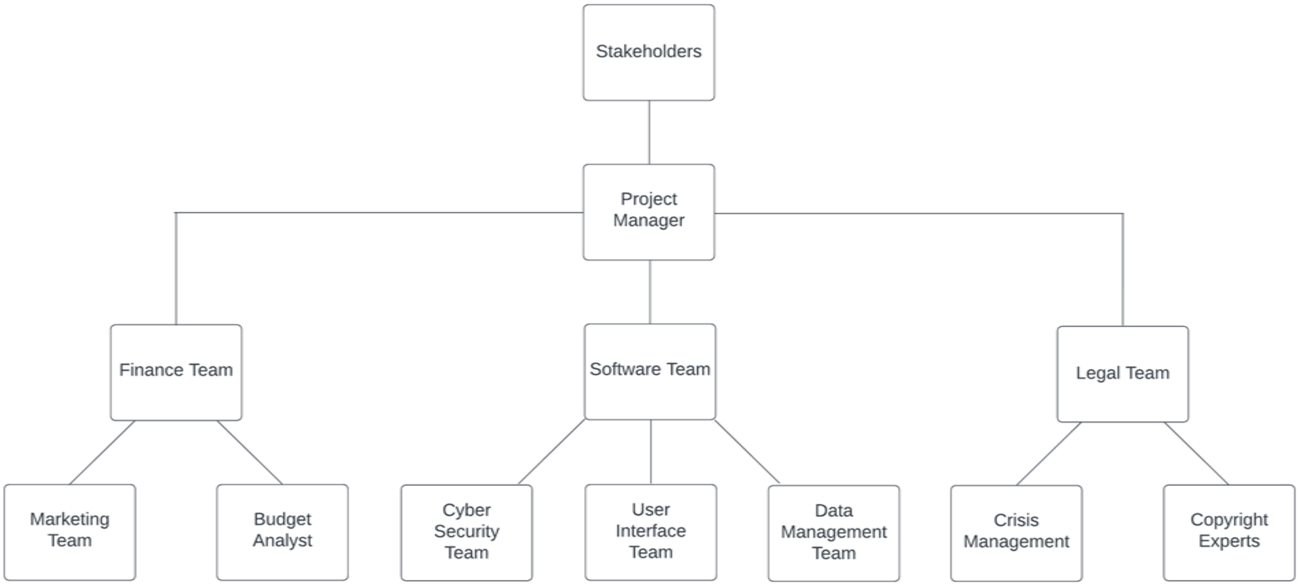
		shown when user open story review mode		
1.2.3.6	Task 6	A copyright notice is sent once a month or when new region is detected		
1.3	**Network Communication**	Have a successful network safety design	Cyber Security Team	162, 000
1.3.1	Requirements Analysis	Have secure network communication	Muhammad Ghazi	
1.3.1.1	Task 1	Decentralize media		
1.3.1.2	Task 2	Encryption		
1.3.1.3	Task 3	Advertisement selection		
1.3.2	Design	Design a secure network		
1.3.2.1	Task 1	Functions as a point-to-point method, where no real media is stored on central databases, instead it is saved on user devices and any new user viewing the shared content has that content delivered to them through other users	Cyber Security Team and Data Management team	30,000
1.3.2.2	Task 2	All media is encrypted when users interact with each other. The media sent through direct messaging, the user profile info, location, and even the packets of decentralized media		

1.3.2.3	Task 3	Advertisers are vetted and any advertisement is checked for potential scams, and trojans		
1.3.3	Code & Unit Test	Successful secure network tested		
1.3.3.1	Task 1	Media is received through parallel downloads of data packets from multiple users		
1.3.3.2	Task 2	System is encrypted where no user info is given through any media posting or through the central servers		
1.3.3.3	Task 3	Advertisements are proven safe for users with no active deception shown or hidden		
1.4	Software Integration	Software Successfully works with the given hardware and app services	Project manager and Software Team	
1.5	Status Review	Review the progress of each iteration	Project Manager	
1.5.1	SRR (System Requirements Review)	Can the app features be implemented with the current technology	Project Manager and Stakeholders	
1.5.2	PDR (Preliminary Design Review)	The design of the features with the given technology	Project manager and Software Team	
1.5.3	CDR (Critical Design Review)	Is the final design up to the specifications of the project	All teams	

1.6	Independent Verification Audit	The project manager must find people with outside perspectives and the stakeholders can be customers reviewing the product	Project Manager and Stakeholders	15,000
1.7	Software Requirements Review Meeting	Software design is checked to see if requirements were met for the given iteration in the project	Software Team and Higher-ups	
1.8	Software Design Review Meeting	The instructions for how the software will be made and designed	Software Team and Higher-ups	
1.9	Consultation	Outside help could be brought in for the more intricate areas	Project Manager and Stakeholders	18,000
1.10	Configuration Management	Will decide what step in the project is next	Project Manager	
1.11	Quality Assurance	Making sure everything is functional and not much will come back to bite them later	Project Manager and Legal Team	25,000
1.12	Documentation	Project Manager has to keep track of everyone's actions and legal team has to check for privileges and repercussions	Project Manager and Legal Team	25,000
1.13	Travel	If a person has to travel the budget is assessed through the manager and stakeholders	Project Manager and Stakeholders	30,000

1.14	Management for all Software	Project manager and software team will make the software in a way that meets all needs	Project manager and Software team	

Organizational Breakdown Structure (OBS)



Basis of Estimate

BOE ID:	WBS: ALL		
Author Name: Steven Jamaica Cervera	**Level 3**	**Period of Performance: 60 hours per leg** **3 people**	
		Total Hours	**1260**

Task Descriptions: (List out all WBS)

<WBS delineation of tasks here>

1.1: User Interface: This incorporates all WBS for User Interface

1.2: Data Management: This incorporates all WBS for Data Management

1.3: Cybersecurity: This incorporates all WBS for Cybersecurity

Basis of Estimate:

Burden Rate = 1.4

1. = 75,000/1.4 = 53,571.40 = lines of code per year
2. = 120,000/1.4 = 85,714.30 = lines of code per year
3. = 192,000/1.4 = 137,143.00 = lines of code per year

1.6 = 15,000/1.4= 10,714.30

1.9 = 18,000/1.4 = 12,857.10

1.11 = 25,000/1.4 = 17,857.10

1.12 = 25,000/1.4 = 17,857.10

1.13 = 30,000/1.4 = 21,428.60

Labor Estimate:

WBS	Task Description	Calculation	Period	Hours
1.1	**User Interface**	**60 * 7**	**420 hours**	[redacted]
1.2	Data Management	60 * 7	420 hours	
1.3	Cybersecurity	60 * 7	420 hours	
				1260

Material Estimate:

WBS	Purchase Description	Calculation	Cost
1.1	**User Interface**	53,571.40	[redacted]
1.2	Data Management	85,714.30	
1.3	Cybersecurity	137,143.00	
1.6	Independent verification	10,714.30	
1.9	Consultation	12,857.10	

1.11	Quality Assurance	17,857.10	
1.12	Documentation	17,857.10	
1.13	Travel	21,428.60	
			303,571.00

Trips Estimate:

WBS	Trip Description	Calculation	Cost
1.9	**Consultation**	12,857.10	**$XXXX**
1.13	Travel	21,428.60	
			34,285

Rationale/Estimating Methodology:

All prices for numbers come from a calculation of average salaries in Texas, level of expertise of software developers, and usually cost of business over the course of a year. The initial budget was very generous in amount that was going to be received and spent. The prices came from a simple calculation of burden rate and total for the year, there could be errors, but future planning is done repeatedly over the course of the project and the budget gets reassessed.

Schedule

This is the overall timeline of the software development, note this is just for the milestones, any meeting with stakeholders and financial and legal teams can take place before the due dates. Testing can be done at any time to combine all software parts from teams. A final trial of the app and its software will be done once all pieces have been put together. This schedule was made with having a working prototype in a minimal time frame.

Snapster™

Timeline of Software Development

First Leg

Name	Subitems	Person	Status	Date
Basic UI	Test	Steven Jamaica	Working on it	2023-03-01
Media Capture and Storage	Test	Manushi Parajuli	Working on it	2023-03-02
Messaging encryption 1st check	Test	Muhammad Ghazi	Working on it	2023-03-04
				2023-03-01 to 2023-03-04

Second Leg

Name	Subitems	Person	Status	Date
Typeface choice	Test	Steven Jamaica	Working on it	2023-03-08
Filter Implementation	Test	Manushi Parajuli	Working on it	2023-03-09
Messaging Encryption 2nd check	Test	Muhammad Ghazi	Working on it	2023-03-10
				2023-03-08 to 2023-03-10

Third Leg

Name	Subitems	Person	Status	Date
Profile Settings	Test	Steven Jamaica	Working on it	2023-03-15

Cache download	Test	Manushi Parajuli	Working on it	2023-03-16
Messaging Encryption Final Check	Test	Muhammad Ghazi	Working on it	2023-03-17
				2023-03-15 to 2023-03-17

Fourth Leg

Name	Subitems	Person	Status	Date
Privacy Settings	Test	Steven Jamaica	Working on it	2023-03-22
Payment Acceptance	Test	Manushi Parajuli	Working on it	2023-03-23
Point to Point Method 1st check	Test	Muhammad Ghazi	Working on it	2023-03-24
				2023-03-22 to 2023-03-24

Fifth Leg

Name	Subitems	Person	Status	Date
Help Settings	Test	Steven Jamaica	Working on it	2023-03-29
Copyright notice implementation	Test	Manushi Parajuli	Working on it	2023-03-30
Point to Point Method 2nd Check	Test	Muhammad Ghazi	Working on it	2023-03-31
				2023-03-29 to 2023-03-31

Sixth Leg

Name	Subitems	Person	Status	Date
Point to Point Method final check	Test	Everyone	Working on it	2023-04-07
				2023-04-07

Last Leg

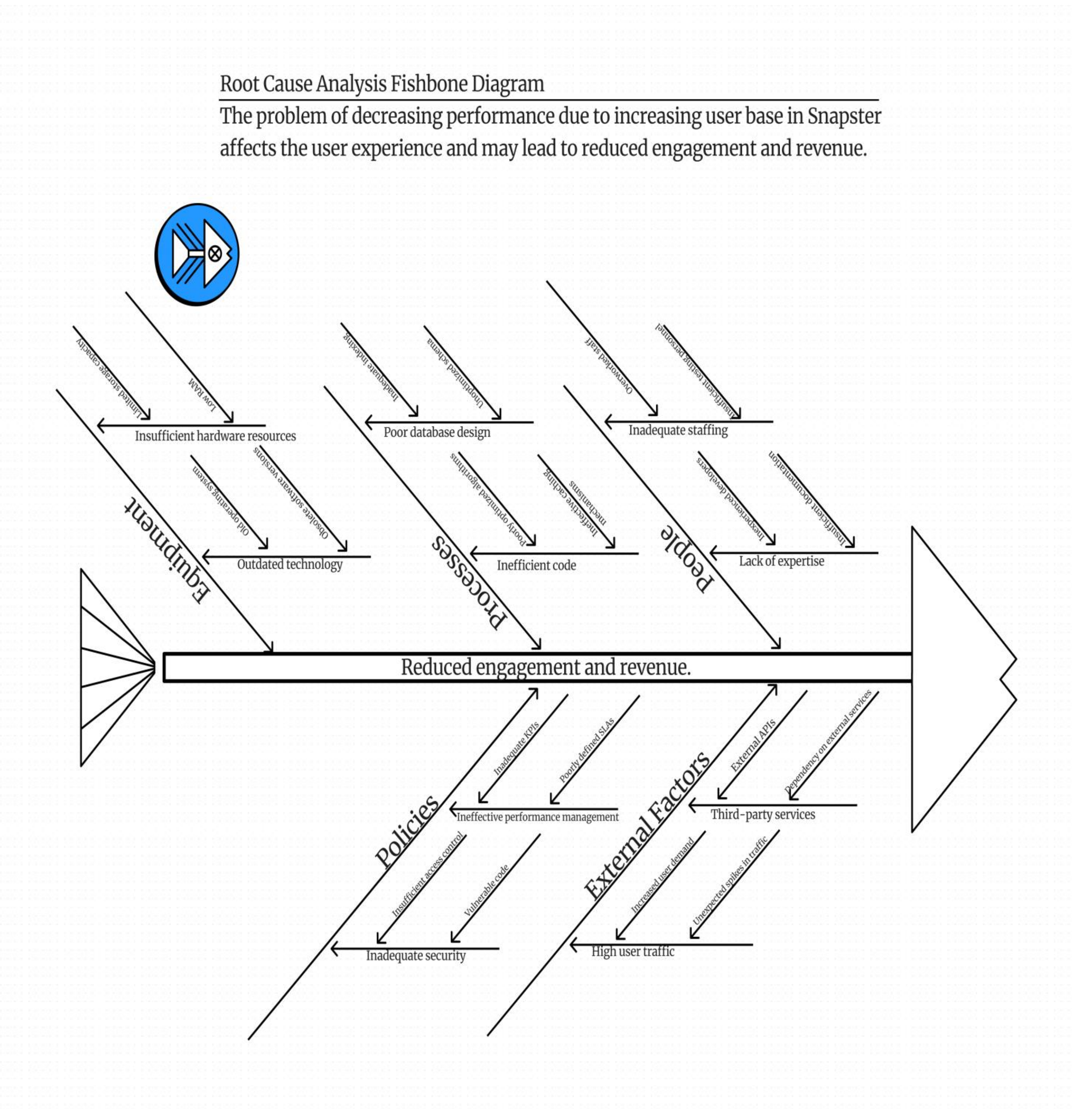
Name	Subitems	Person	Status	Date
Marketing Approval	Test	Muhammad Ghazi	Working on it	2023-04-14
Marketing implamantation	Test	Everyone	Working on it	2023-04-15
				2023-04-14 to 2023-04-15

3. Problem Definition

Problem Statement

Element	Description
The problem of	The problem of decreasing performance due to increasing user base in Snapster affects the user experience and may lead to reduced engagement and revenue.
Affects	Users, Company
the result of which	Users abandoning the platform or negative reviews affecting Snapster's reputation.
benefits of	improved user satisfaction, increased engagement and revenue, and a positive reputation for Snapster.

Root Causes

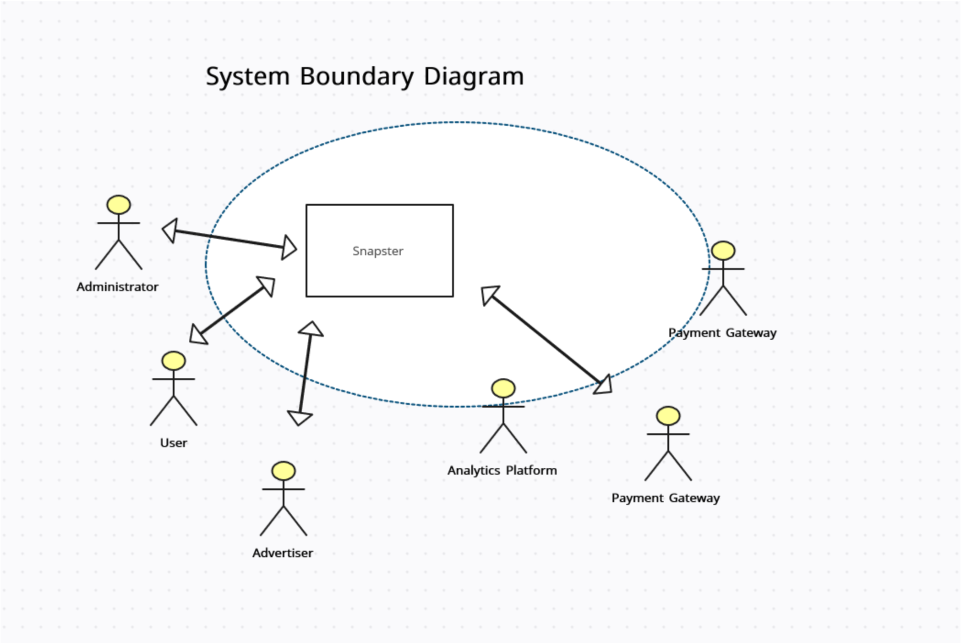

Root Cause Analysis Fishbone Diagram
The problem of decreasing performance due to increasing user base in Snapster affects the user experience and may lead to reduced engagement and revenue.
Equipment
Limited storage capacity
Low RAM
Insufficient hardware resources
Old operating system
Obsolete software versions
Outdated technology
Processes
Inadequate indexing
Unoptimized schema
Poor database design
Poorly optimized algorithms
Ineffective caching mechanisms
Inefficient code
People
Overworked staff
Insufficient testing personnel
Inadequate staffing
Inexperienced developers
Insufficient documentation
Lack of expertise
Reduced engagement and revenue.
Policies
Inadequate KPIs
Poorly defined SLAs
Ineffective performance management
Insufficient access control
Vulnerable code
Inadequate security
External Factors
External APIs
Dependency on external services
Third-party services
Increased user demand
Unexpected spikes in traffic
High user traffic

Stakeholders & Users

Users	Other stakeholders

Businesses	Investors
Salespeople	Advertisers
Multimedia houses	Content creators
Creators	Development Team
Administrator	Financial Manager

System Boundary Diagram

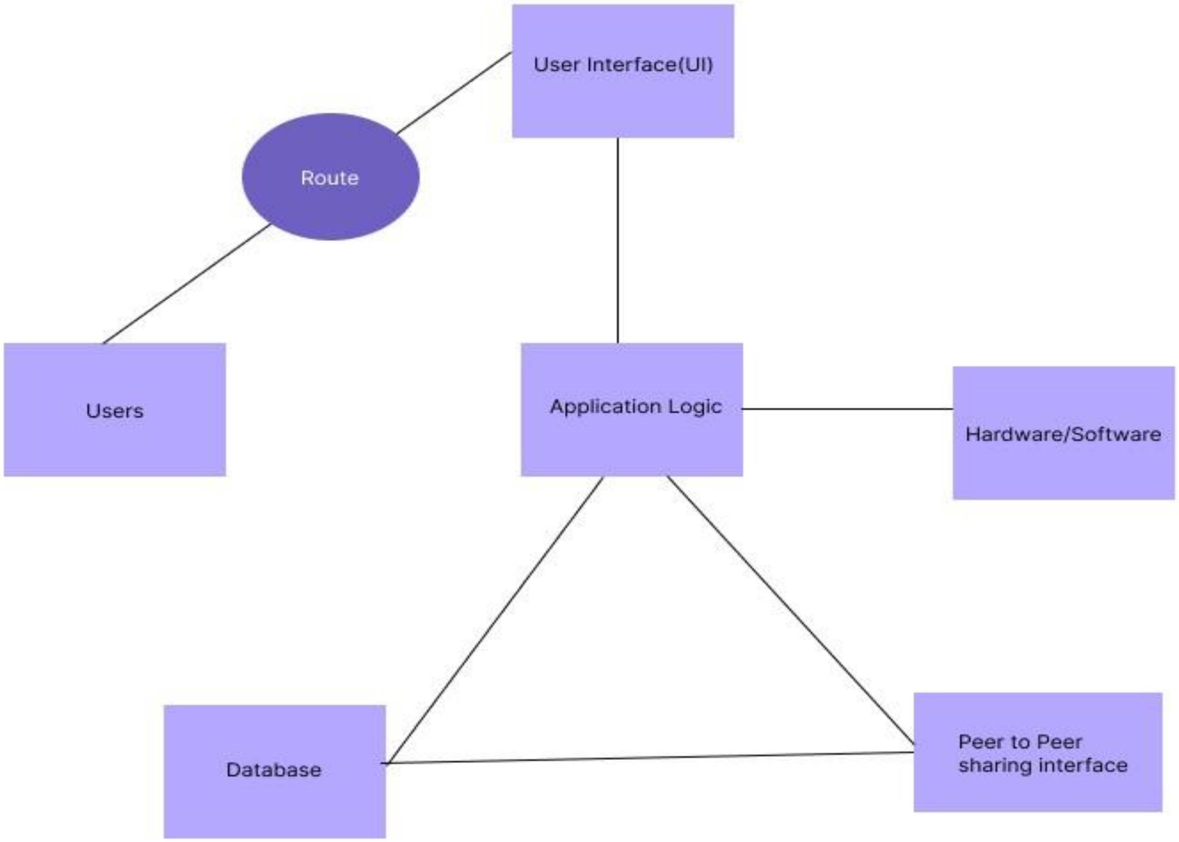


Constraints

Source	Considerations
System	■ Is it to be built on our existing systems? ■ system is scalable, reliable, and secure ■ What O/S and environments must be supported? ■ Data migration compatibility ■ What kind of resources will be required for ongoing maintenance and support?
Environmental	■ Environmental or regulatory constraints? ■ Legal requirements? ■ Security requirements?
Schedule and resources	■ ■ optimizing resource allocation? ■ minimal wastage Usage of outside labor permissible? ■ managing delays? ■ ensuring that project milestones are achieved
Economic	■ What financial or budgetary constraints are applicable? ■ Ways to earn revenue ■ Free or paid app ■ Licensing issues?
Legal	■ Any internal or external political issues that affect potential solutions? ■ compliance with data privacy laws? ■ compliance with intellectual property rights? ■ compliance with industry-specific regulations?

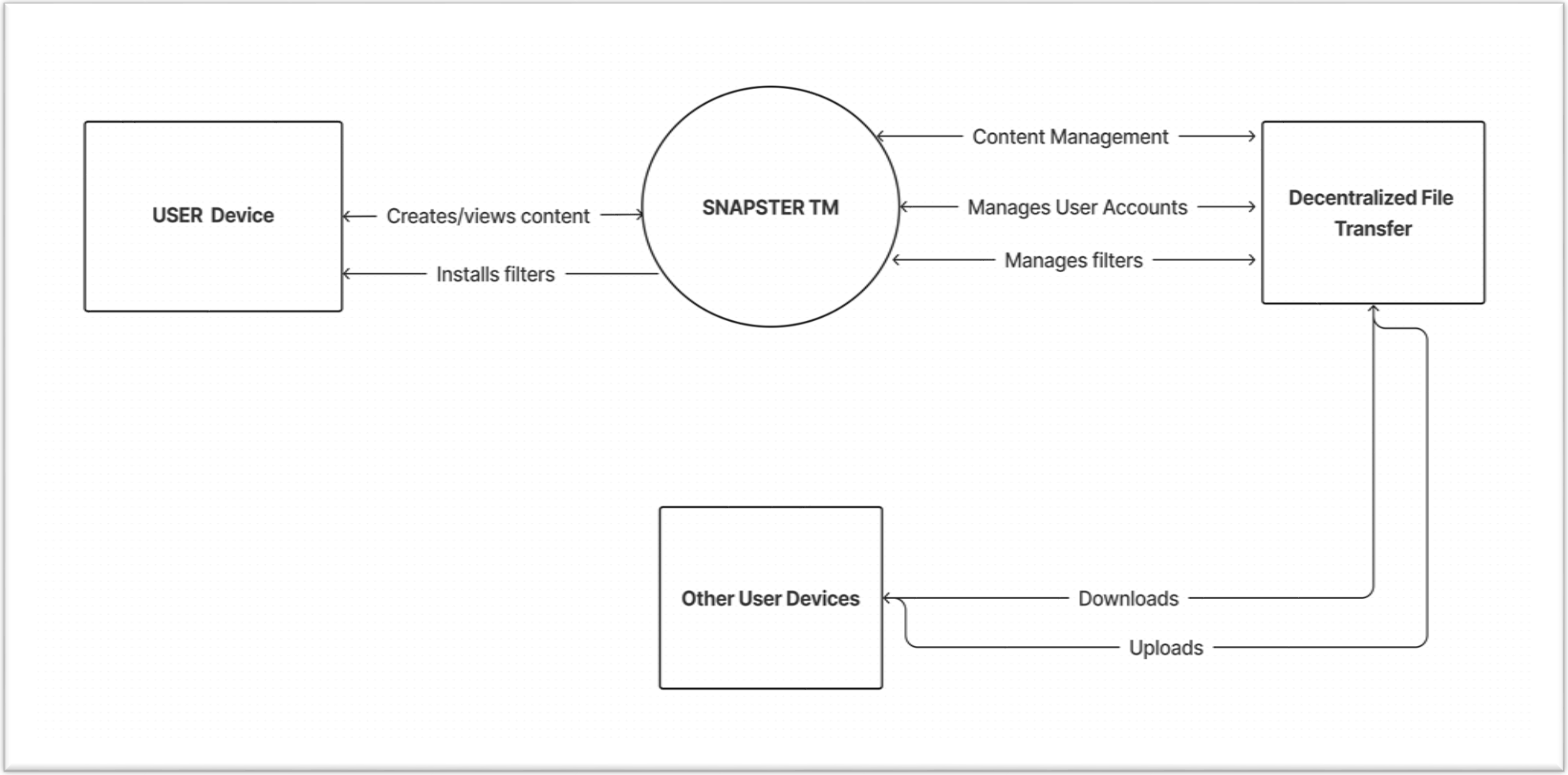
Technical	■ Restricted in any of the app stores? ■ Limited to only new OS? ■ Prohibited from any modern technologies? ■ Licensed plugins?

4. Frame Diagram



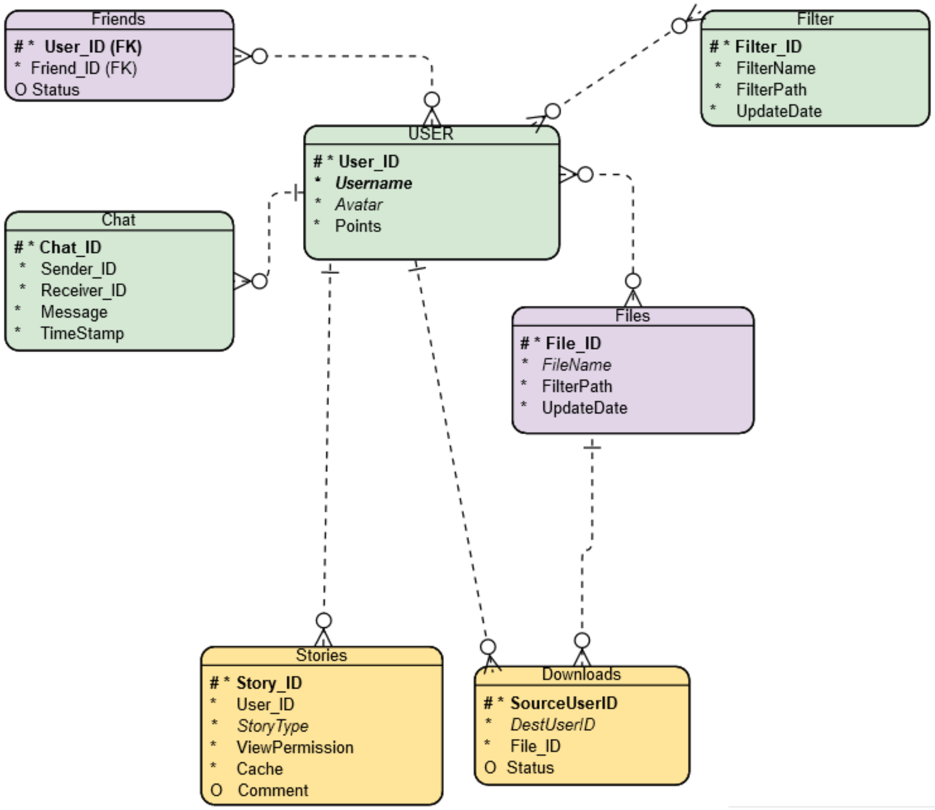
Frame Diagram Snapster

5. Context Diagram



6. Entity Relationship Diagram

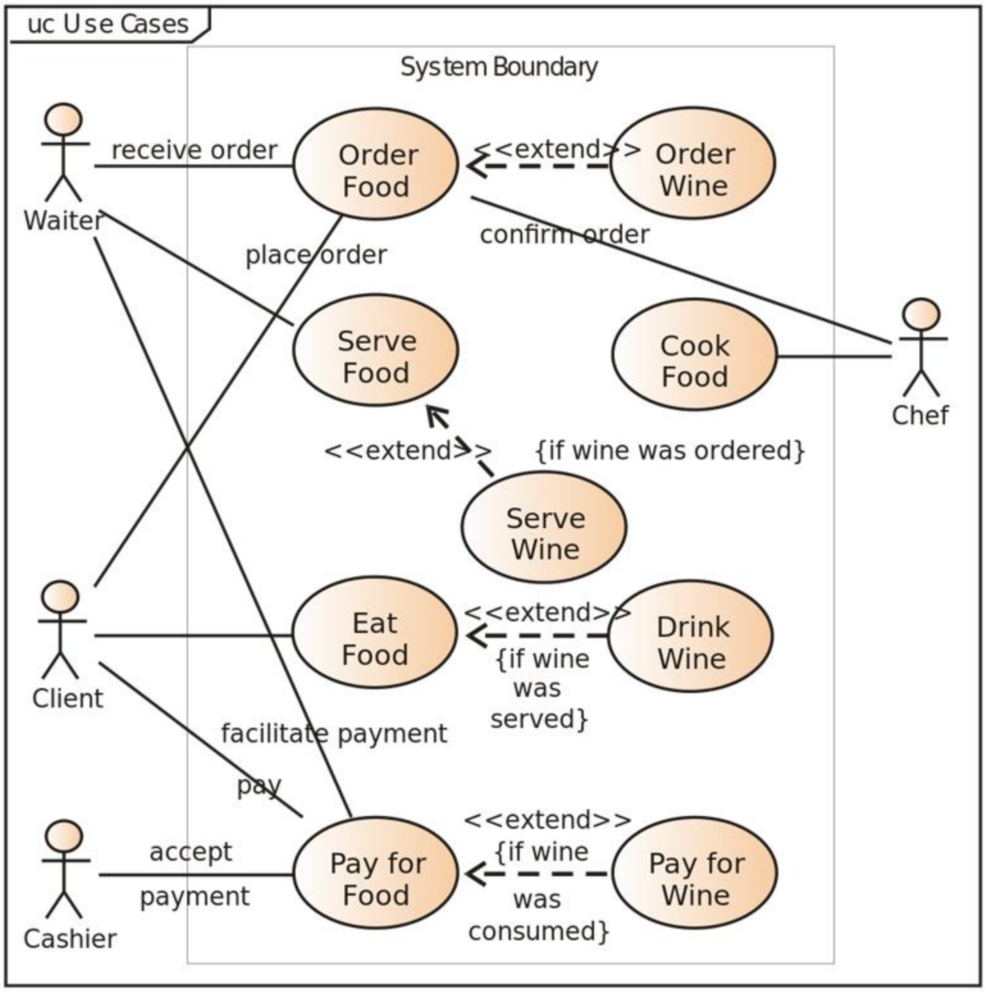
Barker's Notation - ERD (W4RCH4RK3R)



7. Use Cases

Use Case Diagrams

<Insert overall use case diagram, feel free to break it out into multiple diagrams if it starts to become cluttered>



Use Cases

UC ID	UC Name
UC1	Pay for Food
UC1.1	Pay for Wine
UC2	Eat Food
UC2.1	Drink Wine

UC3	Serve Food
UC3.1	Serve Wine
UC4	Cook Food
UC5	Order Food
UC5.1	Order Wine

Normal Scenarios

Use Case: Pay For Food
ID: UC1
Actors: <Actors>
Pre-conditions:
Flow of Events: 1. 2. 3. 4. 5.
Post-conditions: <>

….

Abnormal Scenarios

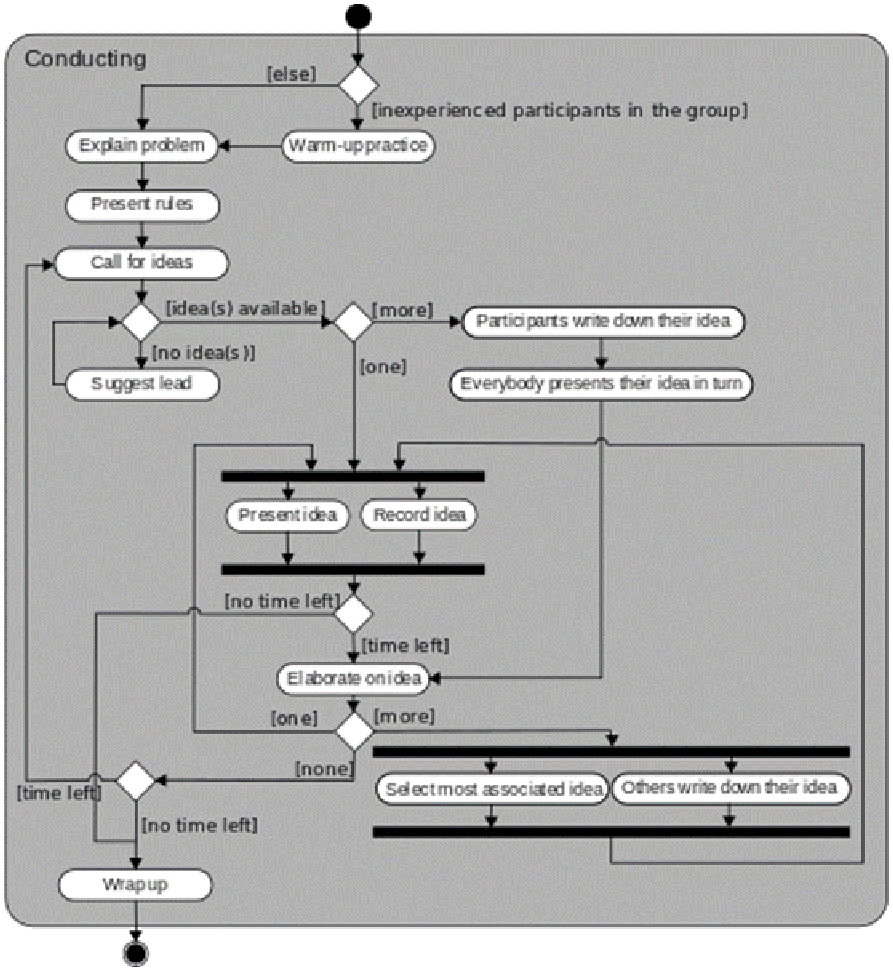
Use Case: Pay For Food
ID: UC1

Actors: <Actors>
Pre-conditions:
Flow of Events: 1. 2. 3. 4. 5.
Post-conditions: <>
Extensions / Alternate Flows:
3a: Invalid credit card: 3a1 The machine displays invalid credit card. 3a2 Return to step 2
Post-conditions: System is ready for credit card processing again
3b: System Timeout: 3b1: 30 seconds expire without insertion of credit card 3b2: System displays “System Timeout” 3b3: Return to step 1.
Post-conditions: System is on main screen ready for a user

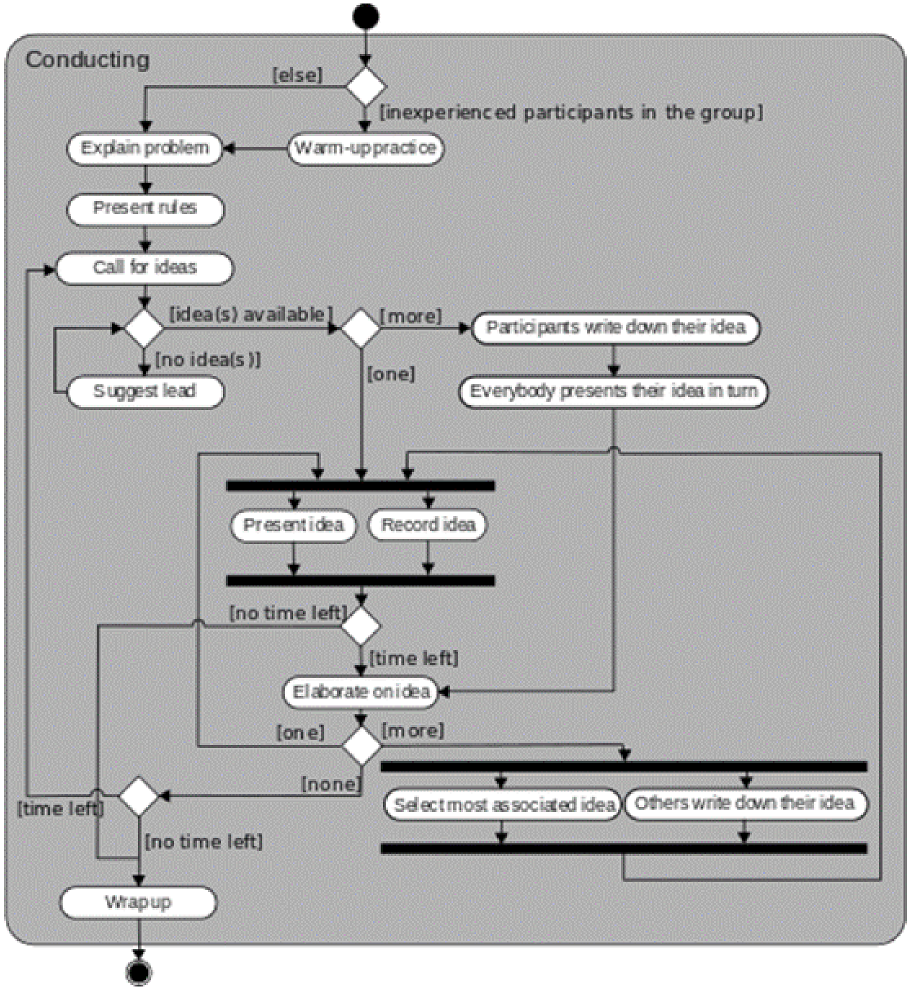
…

8. Activity Diagrams

Normal Scenario



Abnormal Scenario



9. Sequence Diagrams

Normal Scenario

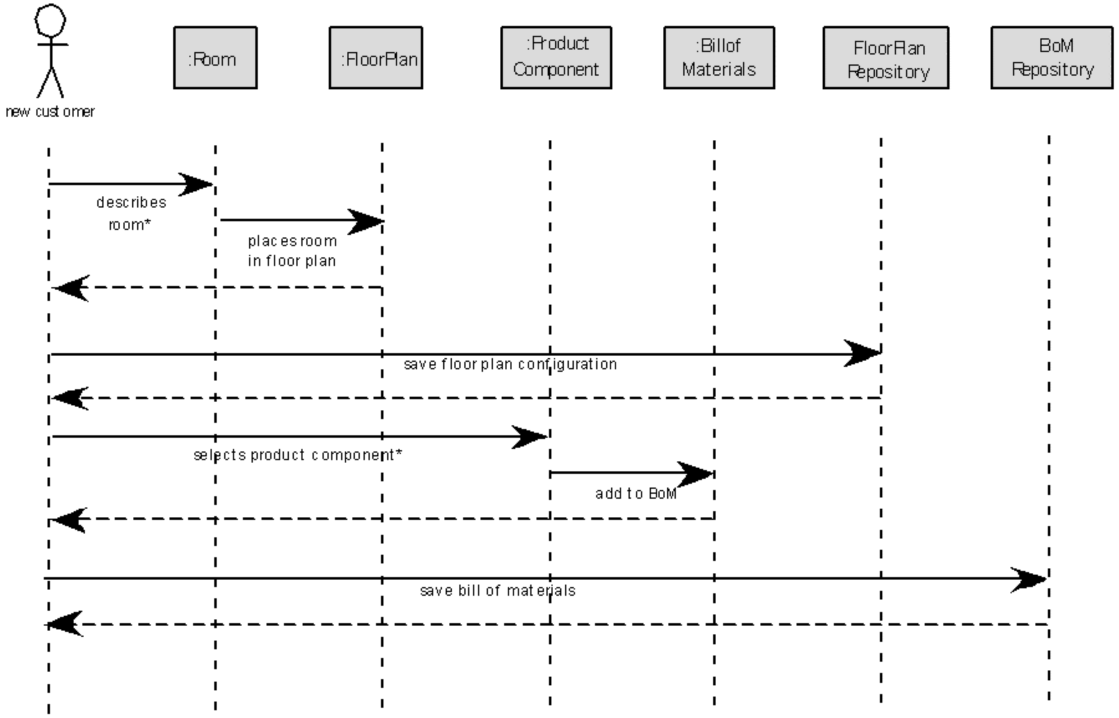


Figure 18.5 Sequence diagram for use-case: *select SafeHome components*

Abnormal Scenario

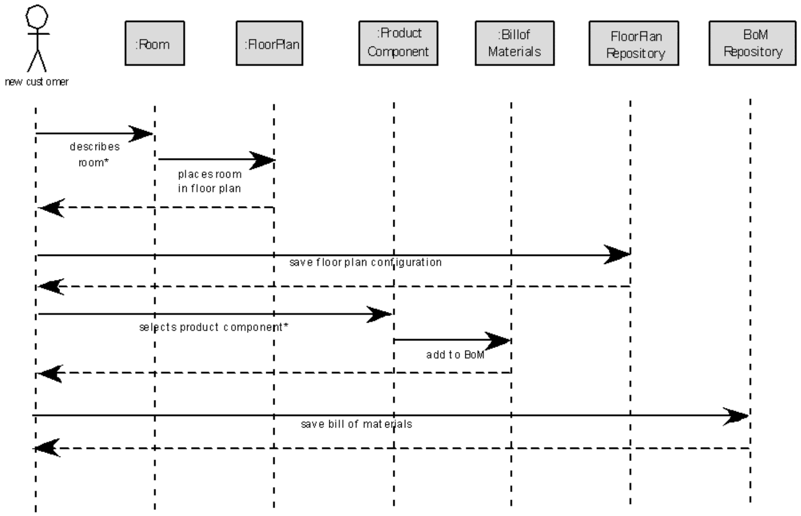
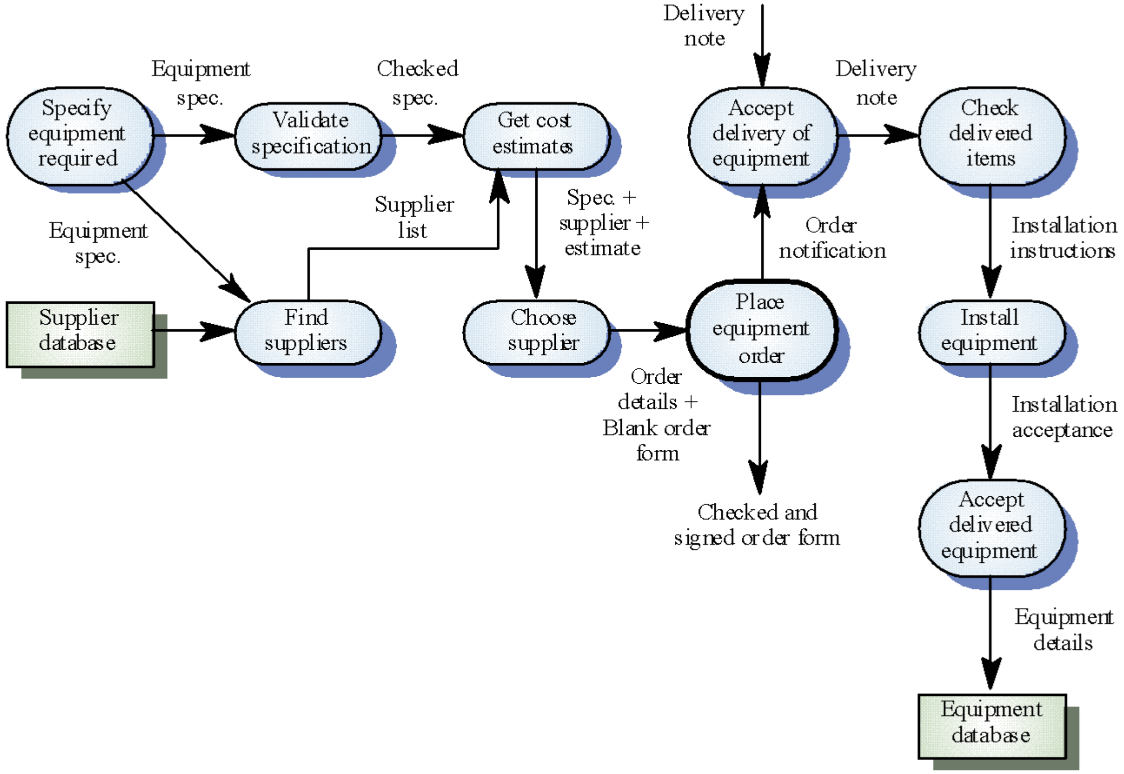


Figure 18.5 Sequence diagram for use-case: *select SafeHome components*

10. Data Flow Diagram



11. Functional Requirements

The following demonstrates iterative requirements development for Mousey. FR 1 and FR1.1 and !.2 may have started with the first 2 statements. "60 seconds has passed" may have been discovered while exploring abnormal scenarios while meching out use cases.

FR1 The game shall implement a turn-based system.

	FR1.1	A Player's Turn shall be completed when a player places a Barrier on the Game Board
	FR1.2	A Player's Turn shall be completed when a player removes a Barrier from the Game Board
	FR1.3	A Player's Turn shall be completed when 60 seconds has passed without an action
	…	…

FR1 The browser shall allow for users to request websites.

	FR1.1	The browser shall allow a user to directly request a website via URL entry
	FR1.2	The browser shall implement click features to navigate to a URL (i.e. hyperlinks)
	…	…

FR1 The editor shall deconflict user input.

	FR1.1	The editor shall lock the area from other user input for up to 20 characters around the user's cursor.
	FR1.2	The editor shall display the conflicted area on other users' screen.
	..	…

12. Nonfunctional Requirements

NF1. The application shall have a crash rate of less than once every 1000 games
NF2. …

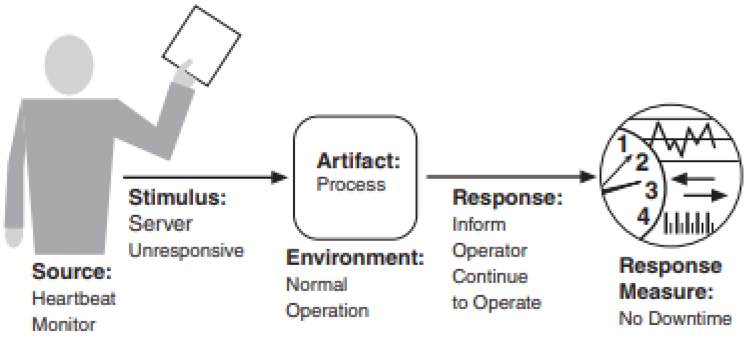
13. Traceability Tables

Use Case	Requirements
UC1	FR19
UC2	FR16
UC3	FR17
UC4	FR16
UC5	FR16
UC6	FR18
UC7	FR13
UC8	FR14
UC9	FR13
UC10	FR1.1
UC11	FR13
UC12	FR15
UC13	FR15
UC14	FR16
UC15	FR13.1
UC16	FR6, FR6.2
UC17	FR6, FR6.1
UC18	FR7, FR8, FR10, FR10.1, FR11, FR12
UC19	FR6
UC20	FR20
UC21	FR19
MUC1	FR15

14. Quality Attribute Scenarios / Tactics

(Availability | Security | Performance | Usability | Interoperability | Modifiability | Testability) Scenario

NFR 1: XXXX



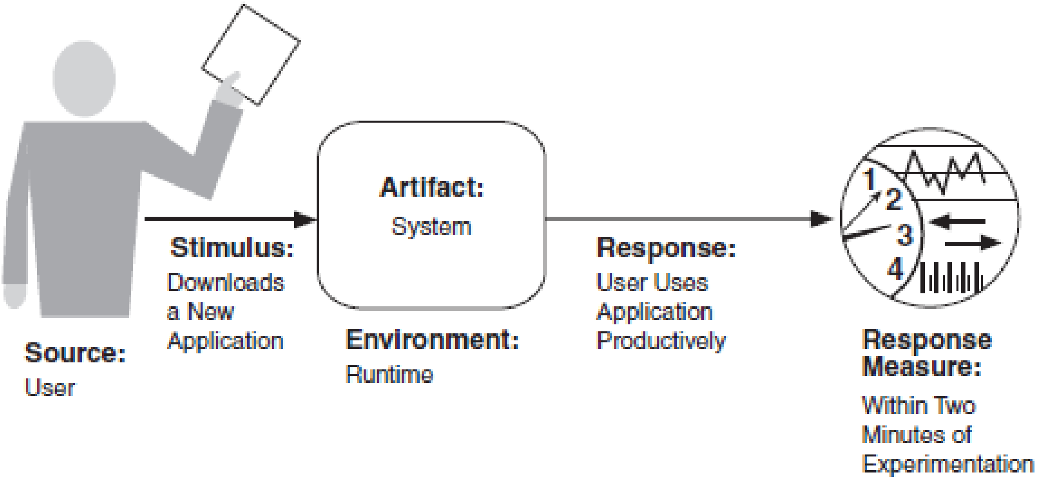
In order to address this requirement, we'll be applying the following tactics:

- XXXX
  - Rationale (By doing this, we'll be able to mitigate this)
  - Detailed Description of Implementation: XXXX
  - Artifacts Affected: Class Diagram, Sequence Diagram
- YYYY
  - Rationale (By doing this, we'll be able to mitigate this)
  - Detailed Description of Implementation: XXX
  - Artifacts Affected: Class Diagram, Sequence Diagram

Actions:

- **Go-Backs: After** reviewing, we've revisited the following design artifacts: "X" diagram.
- **Looking Forward**: Looking forward we will incorporate into the following design artifacts: "X" diarams.
- **Or No Action/Impact:** <Why>

(Availability | Security | Performance | Usability | Interoperability | Modifiability | Testability) Scenario


Source:
User
Stimulus:
Downloads a New Application
Artifact:
System
Environment:
Runtime
Response:
User Uses Application Productively
Response Measure:
Within Two Minutes of Experimentation
1
2
3
4

In order to address this requirement, we'll be applying the following tactics:

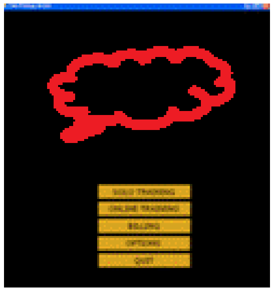
- XXXX
  - Rationale (By doing this, we'll be able to mitigate this)
  - Detailed Description of Implementation: XXXX
  - Artifacts Affected: Class Diagram, Sequence Diagram
- YYYY
  - Rationale (By doing this, we'll be able to mitigate this)
  - Detailed Description of Implementation: XXX
  - Artifacts Affected: Class Diagram, Sequence Diagram

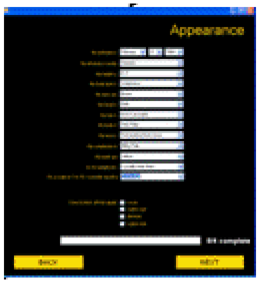
Actions:

- **Go-Backs: After** reviewing, we've revisited the following design artifacts: "X" diagram.
- **Looking Forward**: Looking forward we will incorporate into the following design artifacts: "X" diarams.
- **Or No Action/Impact:** <Why>

15. User Interface Design

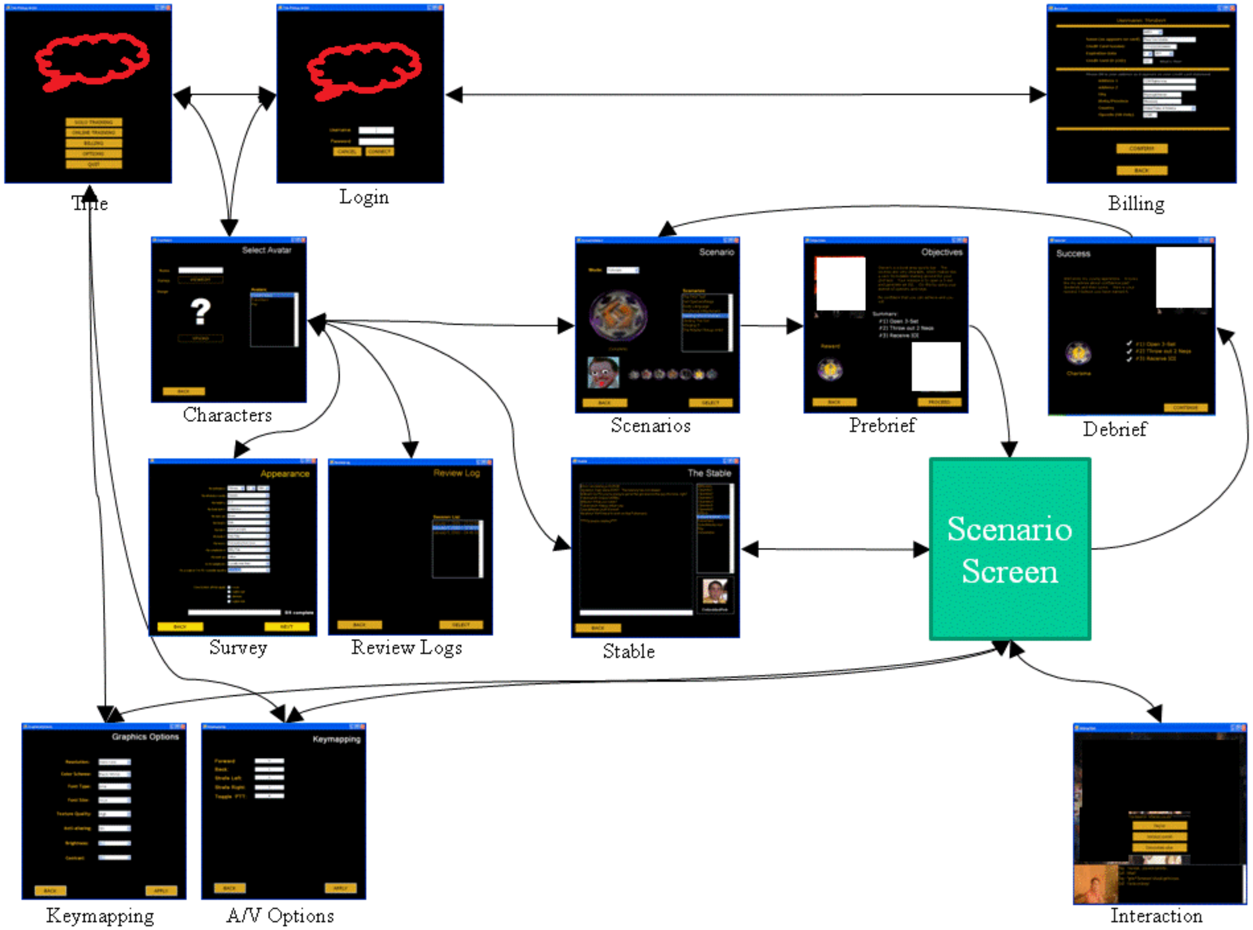
Storyboards



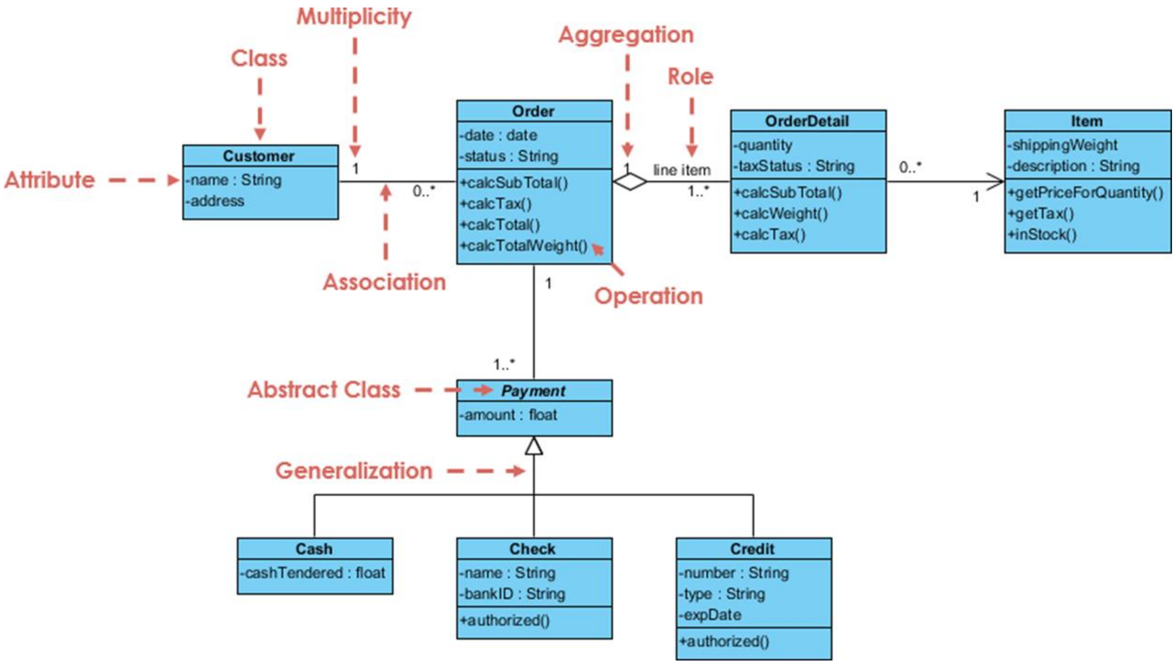


…

Link Analysis



16. Class Diagram



17. Config Management Overview & QA Metrics

Config Management

<Discuss config management>

QA Metrics

<Discuss QA Metrics>

18. System Level Test Plan

Test Case: XXXX

Goal:

Impacted Modules:

Pre-Conditions/Dependencies:

1. AAA
2.

Test Sequence:

Step	Expected Result	Actual Results
1. XXX		
2. BBB		

Final Conditions:

Requirement	Use Cases	Test Case
FR1		
FR1.1	UC10	TC1, TC2
FR2		TC1
FR3		TC1
FR4		TC1
FR5		TC1
FR6	UC16, UC17, UC19	TC3, TC4, TC5
FR6.1	UC17	TC3
FR6.2	UC16	TC4
FR7	UC18	TC3
FR8	UC18	TC3, TC4, TC5
FR9		TC1
FR10	UC18	TC6
FR10.1	UC18	TC6
FR10.2		TC6
FR10.3		
FR10.4		
FR11	UC18	TC7
FR12	UC18	TC8
FR13	UC7, UC9, UC11	TC1
FR13.1	UC15	TC2
FR14	UC8	TC9
FR15	UC12, MUC1, UC13	TC10
FR15.1	UC12	TC10
FR16	UC2, UC4, UC5, UC14	TC11, TC12, TC13
FR17	UC3	TC14
FR18	UC6	TC15
FR19	UC1, UC21	TC16
FR20	UC20	

19. Glossary

AI Player	A Player controlled by the system

20. Bibliography

- *Lausen, Soren. Software Requirements Styles and Techniques. Great Britan: Pearson Education Limited, 2002.*
- *Welcome to monday.com | a new way of working*. (n.d.). https://stevej182.monday.com/auth/login_monday/email_password