



UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

FUNCTIONAL SPECIFICATION

PROJECT NAME: ARCANE ARCADE

CLIENT: TONY VD LINDEN

TEAM NAME: TERABITES

TEAM MEMBERS

NG Maluleke	D Mulugisi	C Nel	LE Tom
13229908	13071603	14029368	13325095

September 12, 2016

Contents

1	Introduction	2
2	Vision	2
3	Functional Requirements and Application Design	3
3.1	Use Case Prioritization	3
3.1.1	Critical	3
3.1.2	Important	3
3.1.3	Nice-To-Have	3
3.2	System Modules	4
3.2.1	User Authentication Module	5
3.2.2	User Management Module	6
3.2.3	Challenge Management Module	10
3.2.4	Level Management Module	13
3.2.5	Question Management Module	15
3.2.6	Compiler Module	18
3.2.7	Player Interaction Module	18
4	Domain Model	18
5	Open Issues	19

1 Introduction

The name *Arcane Arcade* references the esoteric language that users will have to use to interact with the system. It also highlights the gamification approach that makes it a fun activity. The platform will be used to test prospective employees in a fun gamified way. It will indicate in which programming area the user is likely to be better suited, by looking at how the user completed the challenges, such as whether the user used any hints and the badges that the user earned.

2 Vision

The vision is to provide a fun and easily accessible platform that can be used to gauge the programming aptitude and capabilities of existing or future software developers using a custom esoteric programming language.

3 Functional Requirements and Application Design

3.1 Use Case Prioritization

This section outlines the three-level prioritization of the system's required functionality.

3.1.1 Critical

- **User Authentication** - This is critical as it limits the system's access only to people who have been authorized. Only registered users are then able to complete the challenges and/or modify the challenges together with authorizing more users. No one can then access any part of the system unless they have been registered and authorized to request such services.
- **Player Interaction** - Interaction is critical as the whole system's core functionality revolves around it. The players have to be able to view and take challenges that are open to them at that time.
- **Code Compilation** - Code compilation is a critical use case as other instances of Player Interaction revolve around it. Where a Player interacts by typing code the system needs to be able to compile that code.

3.1.2 Important

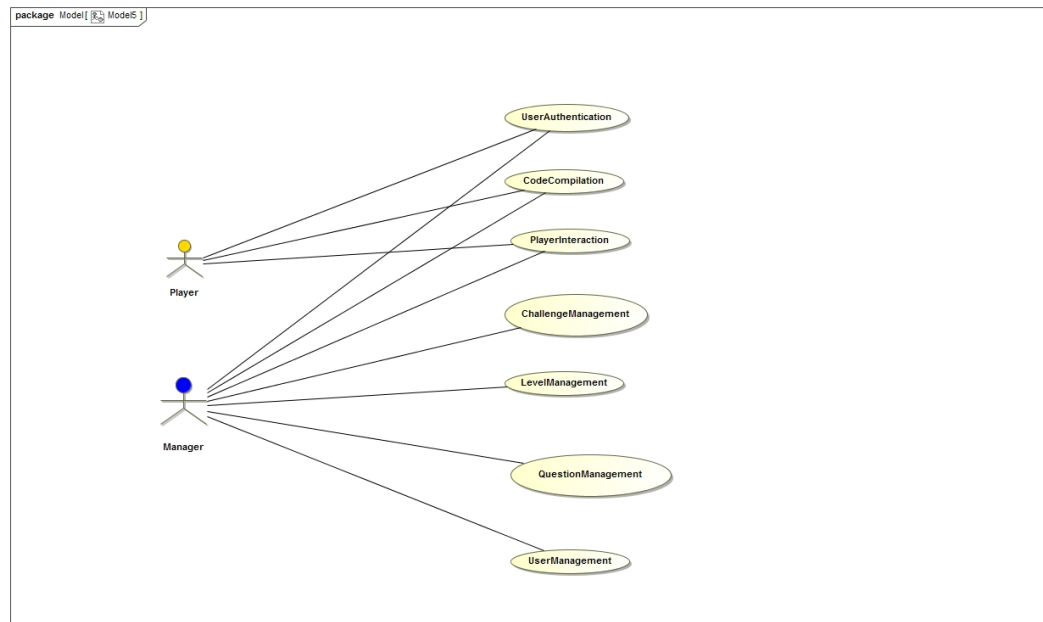
- **Challenge Management** - This is considered important because this is where users who have management privileges will configure the structure of the challenges such as the number of levels and questions. The managers will also be able to set hints and other items related to the challenges.
- **Level Management** - This use case is important and it consists of the functionality of adding and removing levels on challenges and this can only be done by the administrator.
- **Question Management** - This use case is important and it consists of the functionality of adding and removing questions.
- **User Management** - This use case is important as it deals with adding and removing users to the system.

3.1.3 Nice-To-Have

- **Achievements** - This is considered nice to have as it for promoting the game and to get more people to play.
- **Social Media** - This is considered nice to have. It is to showcase in-game achievements on social media. It strengthens competition and markets the game.

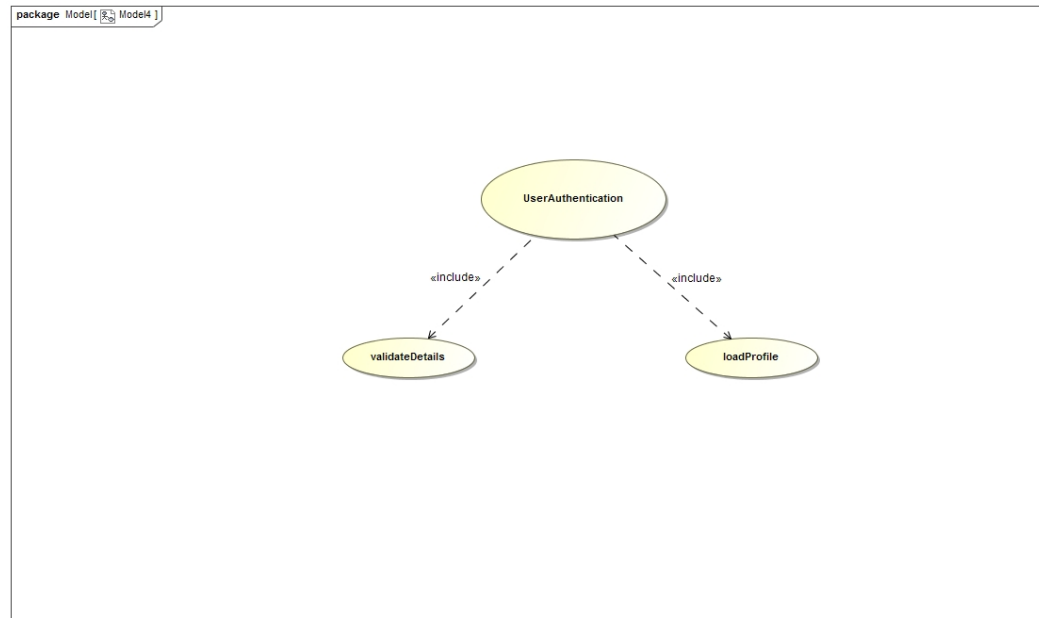
3.2 System Modules

The system has built as a series of separate modules. This approach makes it possible to add more modules at a later stage and to modify one module without interfering with the others. It then allows the system to be pluggable and integrable.



3.2.1 User Authentication Module

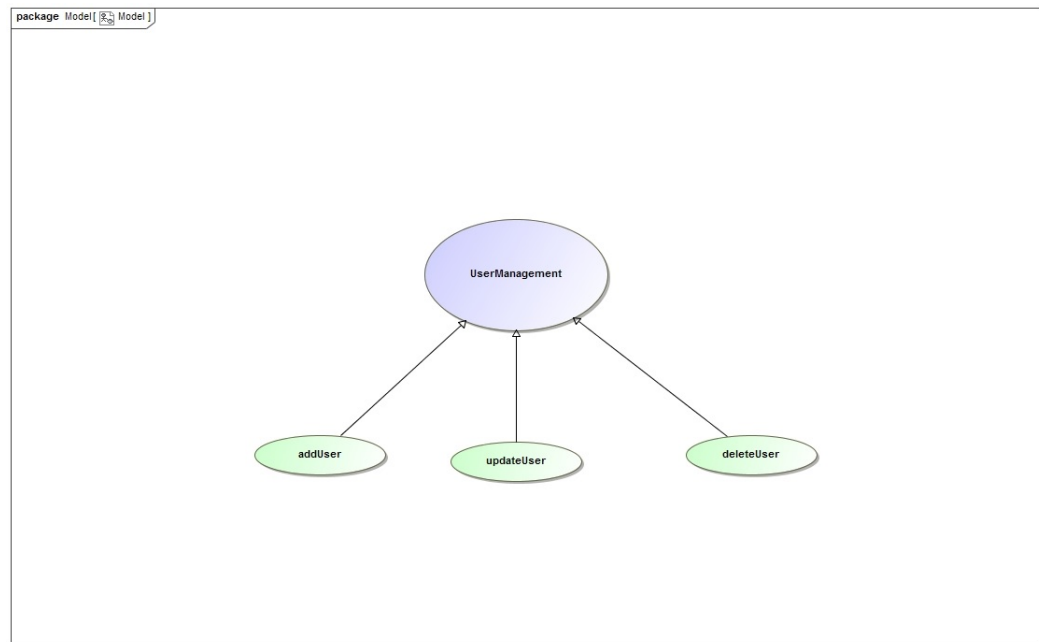
This module provides the functionality which validate the user credential to check if they match up with the access that they request. The most crucial part and main focus of the authentication module is to verify if a user has access to certain pages given the type of user they are. This then improves the system's security and keeps the data confidential as it controls who has access to which functionality.



- **validateDetails** - Any user who uses the system has to login first. The user provides a username and a password which is compared with the credentials that are stored in the system database. If the credentials match then the user is given access to the system.
- **loadProfile** - After a user's credentials have been verified, their profile has to be loaded. This involves gathering data about the game which includes their progress, badges, hints, and credits to buy hints.

3.2.2 User Management Module

This module focuses on the ability to grant and revoke user access to the system. Only the super user has access to the module and it can be used to add, remove and update user accounts.



- **addUser** - The manager has the ability to add more user accounts via the user management page thereby allowing them to access the system. Users are sent an email containing their login credentials after they are added to the system. The service contract and activity diagram for addUser are shown below.

Service Contract

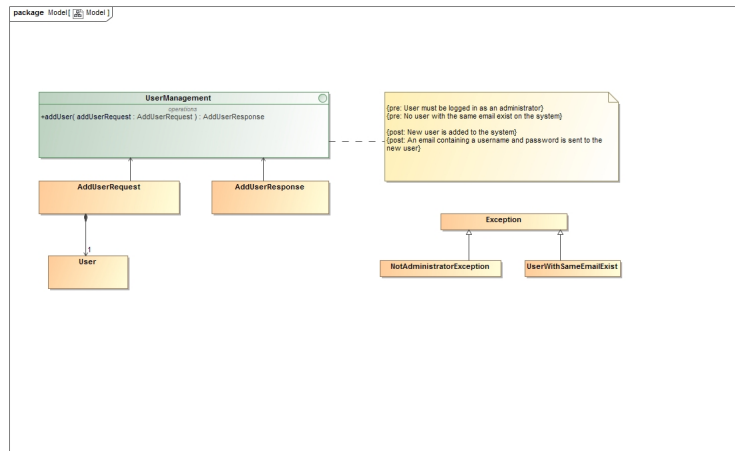


Figure 1: Add new user to the system

Activity diagram

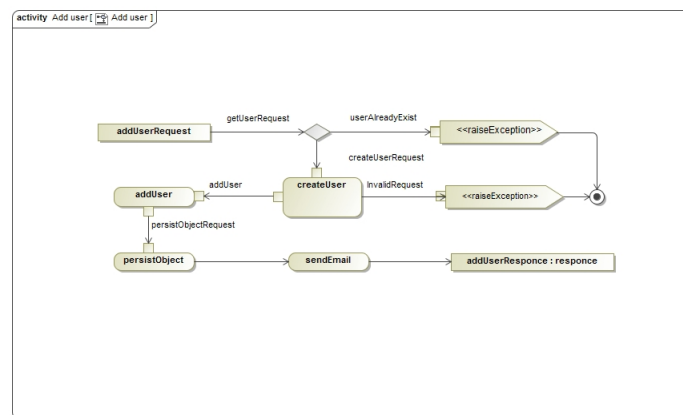


Figure 2: The activity diagram for addUser

- **updateUser** - The updateUser use case falls under User as depicted in the above diagram. The manager has the ability to modify user accounts items via the user management page. The service contract and activity diagram for updateUser are shown below.

Service Contract

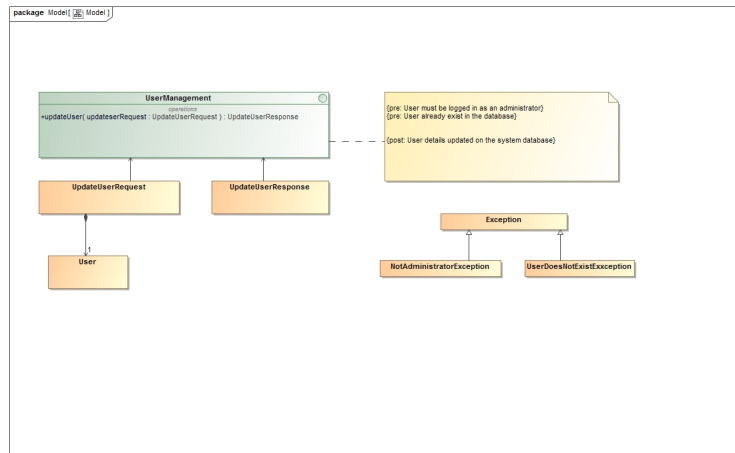
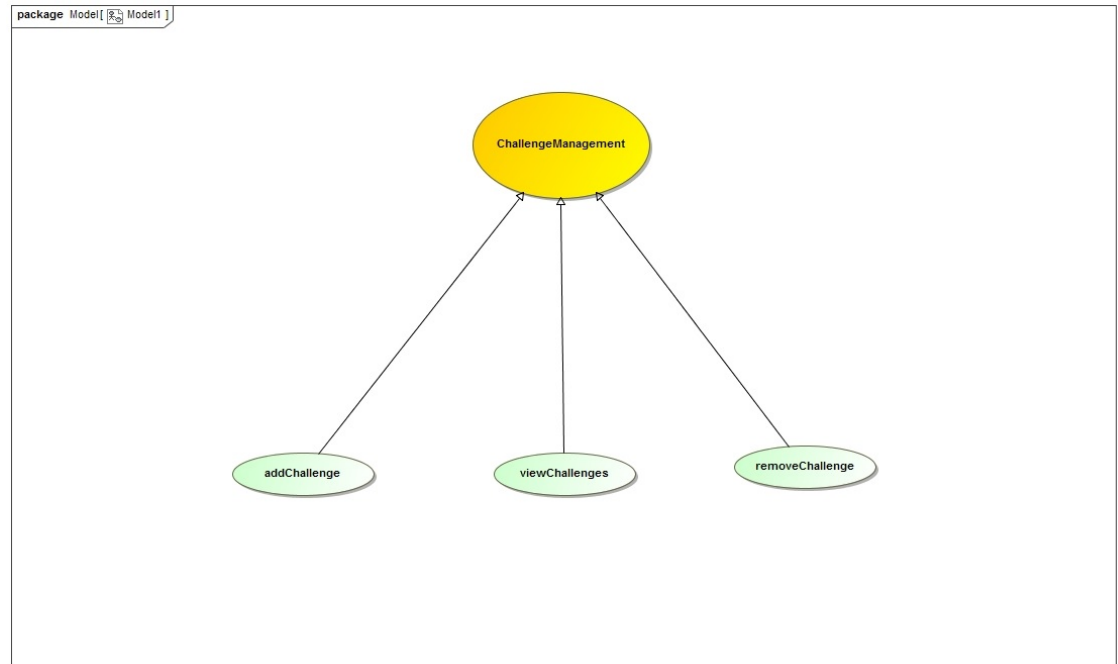


Figure 3: Update User on the system

3.2.3 Challenge Management Module

Only super users(management accounts) have access to this module.It provides the ability to add, remove and modify the structure of challenges. It allows the manager to modify the challenge at any of its three levels which are challenge, level and question. The modifications that the manager can do are add, remove and update any of the three levels of a challenge.



- **addChallenge** - The addChallenge use case falls under Challenge Management module as depicted in the above diagram. The addChallenge submodule allows the administrator to add challenges to the system via the Challenge Management page. The service contract and activity diagram for addChallenge are shown below.

Service Contract

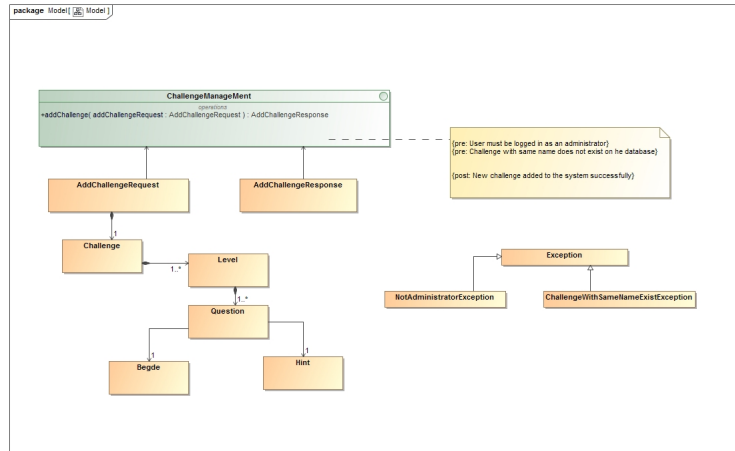


Figure 6: Add challenge to the system

- **removeChallenge** - The removeChallenge use case falls under Challenge Management module as depicted in the above diagram. The removeChallenge submodule allows the administrator to remove challenges from the system via the Challenge Management page. The service contract and activity diagram for removeChallenge are shown below.

Service Contract

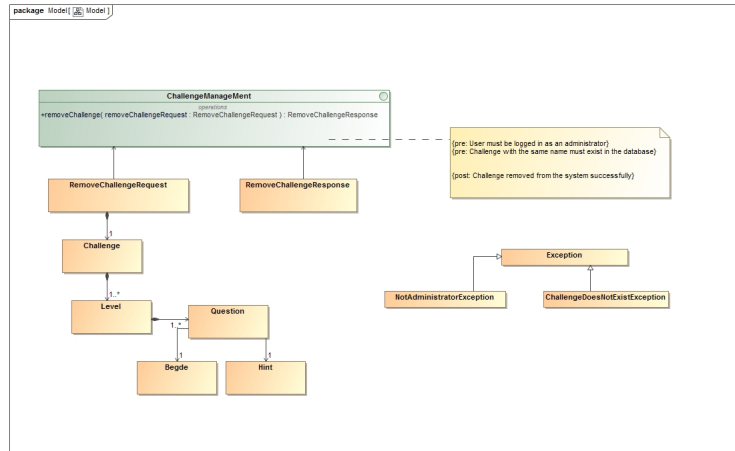


Figure 7: Remove challenge from the system

- **viewChallenges** - The viewChallenges use case falls under Challenge Management module as depicted in the above diagram. The viewChallenges submodule allows the administrator to view challenges that are on the system via the Challenge Management page. The service contract and activity diagram for viewChallenges are shown below.

Service Contract

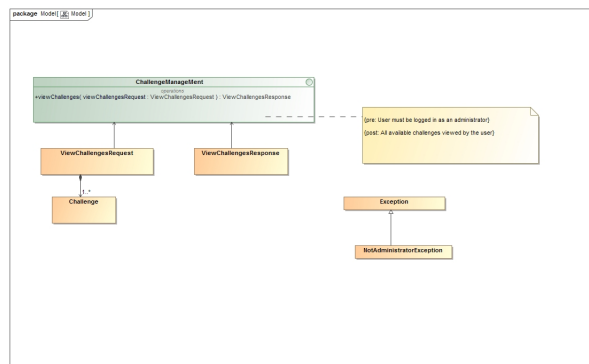


Figure 8: View all the challenges on the system

Activity diagram

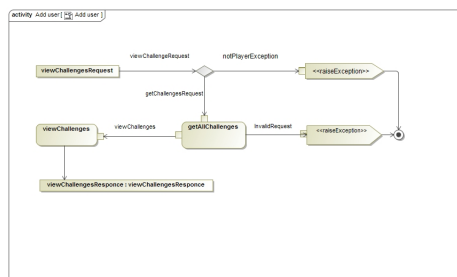
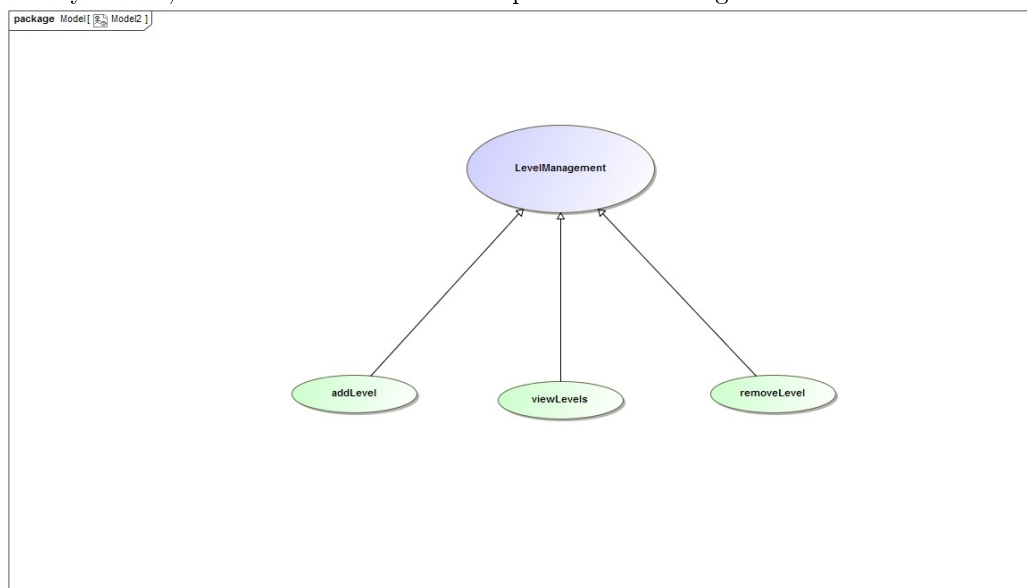


Figure 9: The activity diagram for ViewChallenges

3.2.4 Level Management Module

Only super users(management accounts) have access to this module.It provides the ability to add, remove and view levels of a particular challenge.



- **addLevel** - The addLevel use case falls under Level Management module as depicted in the above diagram. The addLevel submodule allows the administrator to add levels to challenges on the system via the level Management page. The service contract and activity diagram for addLevel are shown below.

Service Contract

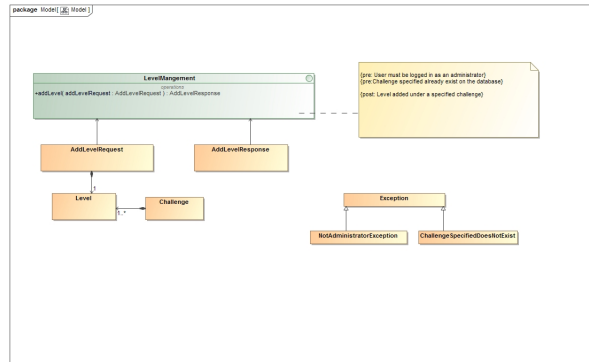


Figure 10: Add level to a challenge

- **removeLevel** - The removeLevel use case falls under Level Management module as depicted in the above diagram. The removeLevel submodule allows the administrator to remove levels from challenges on the system via the level Management page. The service contract and activity diagram for addLevel are shown below.

Service Contract

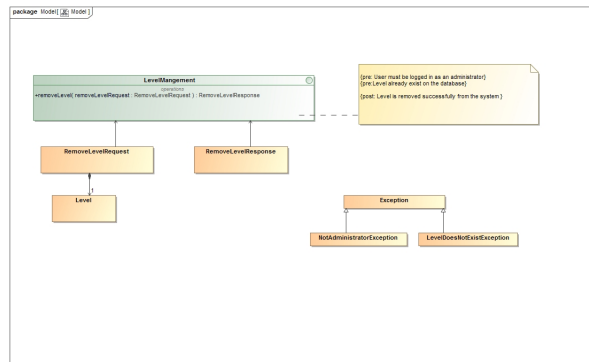
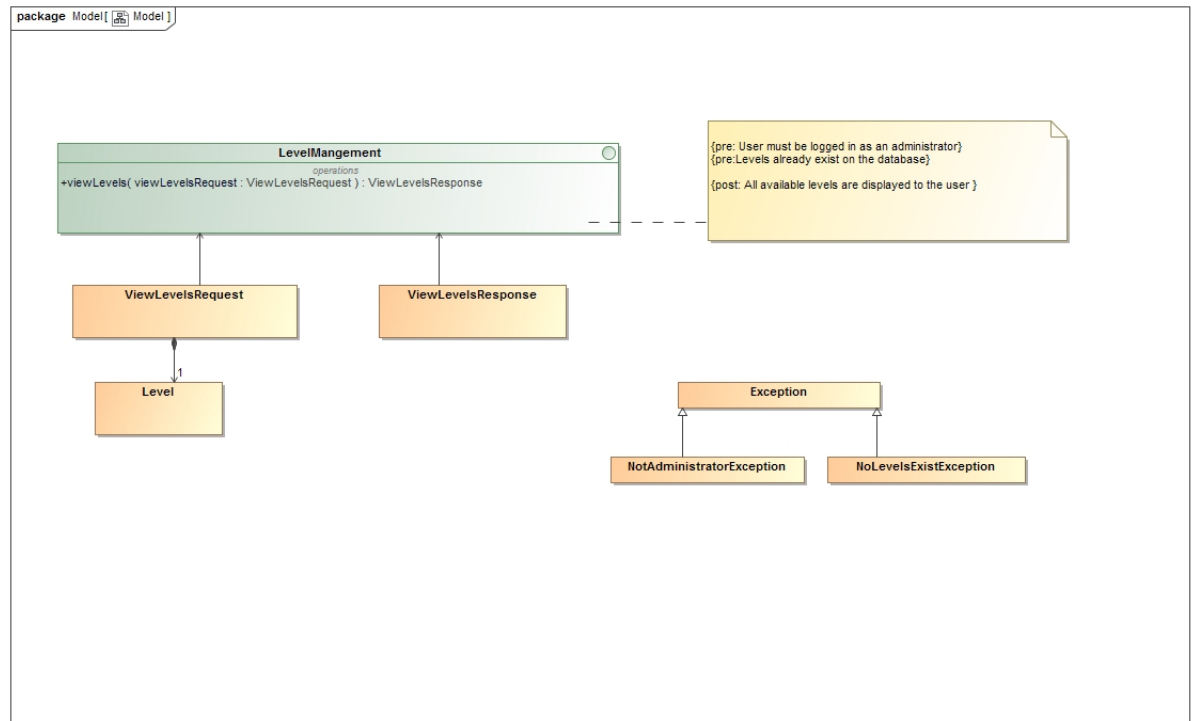


Figure 11: Remove level from a challenge

- **viewLevels** - The viewLevels use case falls under Level Management module as depicted in the above diagram. The viewLevels submodule allows the administrator to view all the levels from challenge on the system via the level Management page. The service contract and activity diagram for viewLevels are shown below.

Service Contract

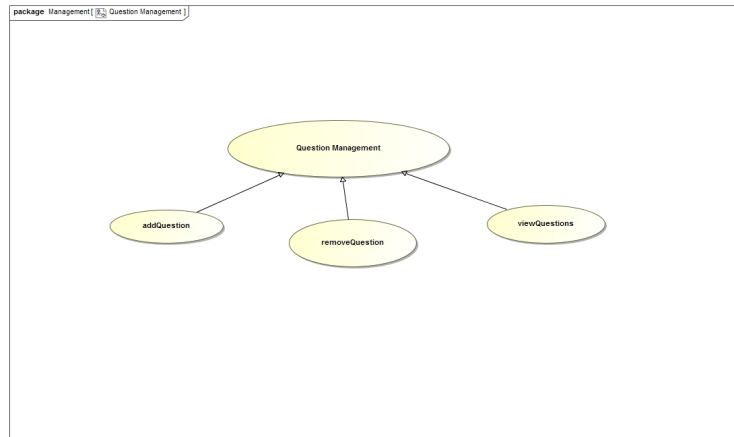


S

Figure 12: View all levels on a challenge

3.2.5 Question Management Module

The Question Management Module can only be used by Managers. It provides the ability to add, remove and modify the structure of Questions. The manager can modify any question, add a new one or delete an existing one. It has the following Services.



- **addQuestion** - The manager has the ability to add questions to an existing Level via the management page. The service contract and activity diagram for addQuestion are shown below.

Service Contract

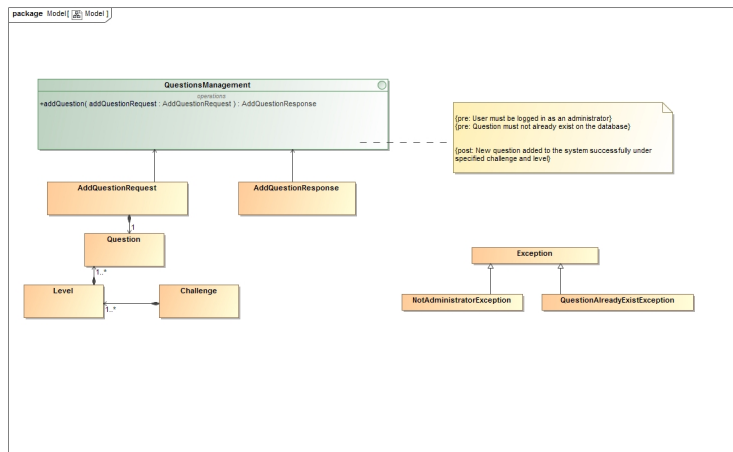


Figure 13: View all levels on a challenge

- **removeQuestions** - The manager has the ability to remove Questions though the management portal. This questions belong to levels which in turn belong to challenges. The service contract and activity diagram for removeQuestion are shown below.

Service Contract

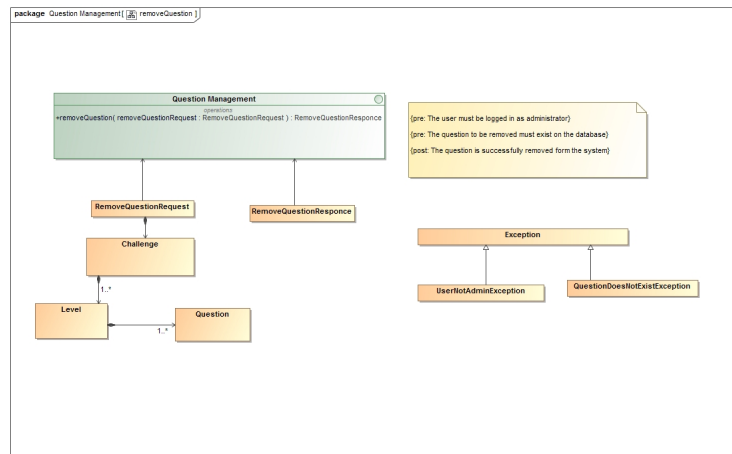


Figure 14: View all levels on a challenge

- **viewQuestions** - The viewQuestions use case provides the manager the ability to get a list of all Questions from all Levels on the system. The service contract and activity diagram for viewQuestions are as follows.

Service Contract

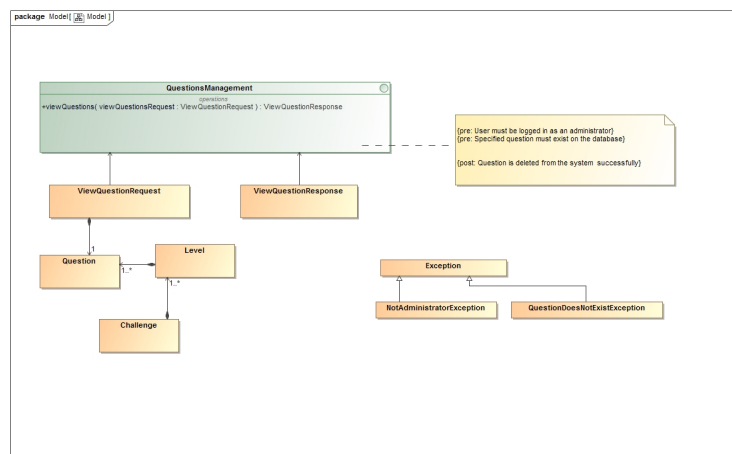


Figure 15: View all levels on a challenge

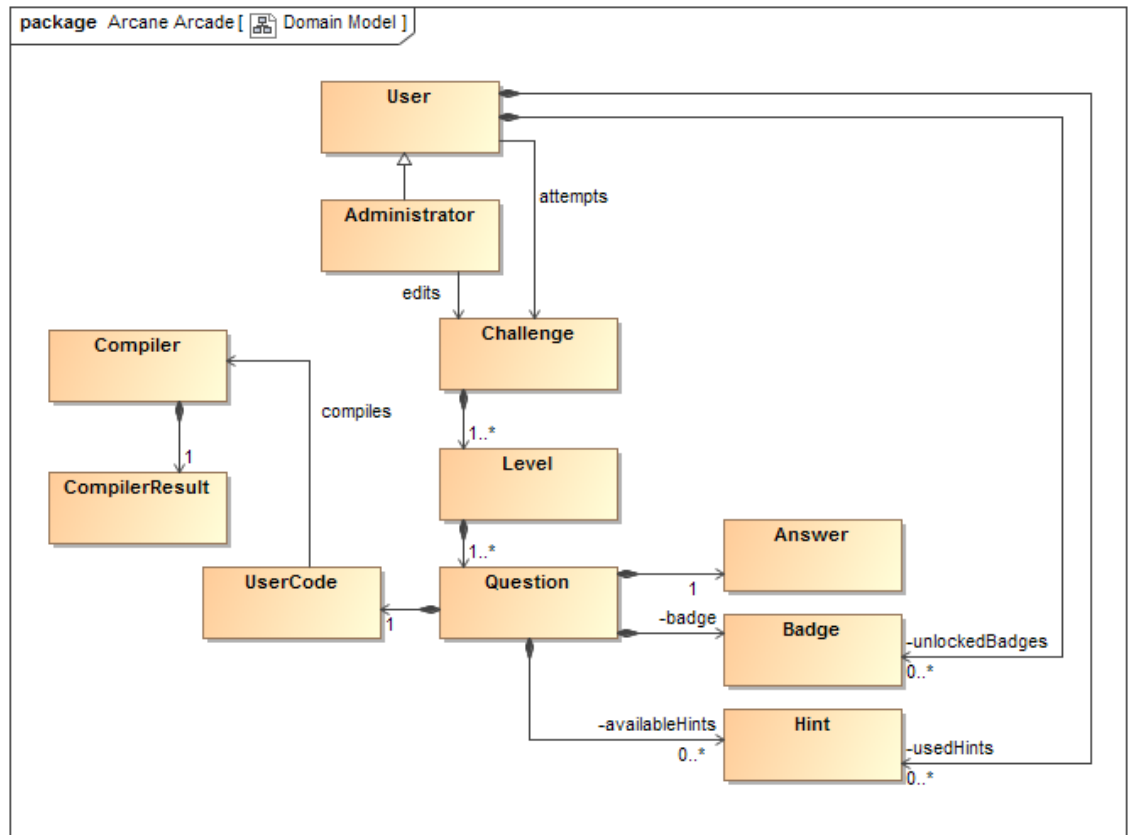
3.2.6 Compiler Module

This module is the backbone of the system. It is responsible for everything that has to do with the user's code. It compiles, executes and displays output that has been generated by the user code.

3.2.7 Player Interaction Module

This module houses the system's main functionality. The main reason of having the system is to allow the users to interact with the system and complete the offered challenges. This use case diagram indicates the functionality around interacting with the system as a player.

4 Domain Model



5 Open Issues