

Project Scope Statement

RAIDDON

Created by:
Farah Gauduin
farah.gauduin@gmail.com

ABSTRACT

World of Warcraft addons can make everyday activities like questing and trading much easier. Blizzard's critically acclaimed MMORPG has a habit of hiding a lot of useful data it collects, and picking up a mod that offers a peek behind the curtain can be a literal gamechanger when going up against deadly bosses. Of course, the best WoW addons aren't just aimed at avid raiders. Some are more focused on utility, freshening up the multiplayer game's geriatric UI, while others help track material nodes for those all-important professions.

Raididon is a world of warcraft addon/website aiming to provide helpful features for world of warcraft classic players. It mainly offers services that many players may find useful and that do not exist within the default game interface. Vous voulez insérer une image à partir de vos fichiers ou ajouter une forme, une zone de texte ou un tableau ? Procédez comme suit : Sous l'onglet Insertion du ruban, appuyez simplement sur l'option souhaitée.



Chapter 1.

Introduction

- The foundations of the project
- Objectives of the project
- Raiddon services
- User Issues
- Raiddon Logo



1. The Foundations of the project

Raiddon is a project that fosters all the development and conceptual understandings acquired throughout the ISIKA 10-month educational program. The key points that the project should satisfy are detailed in the following sections.

1.1 Context and architecture of the project

Raiddon requires the development of a modular application that amounts to drawing a parallel with an assembly of different blocks, each fulfilling very specific functions and having as few outward dependencies as possible. This allows to have several variants of the same application and therefore to be able in a minimum of time to present the same application with different functionalities. In other words, the project is broken down into multiple micro-services, each having its own logic, that are as autonomous as possible; therefore, guaranteeing Raiddon better maintainability and scalability.

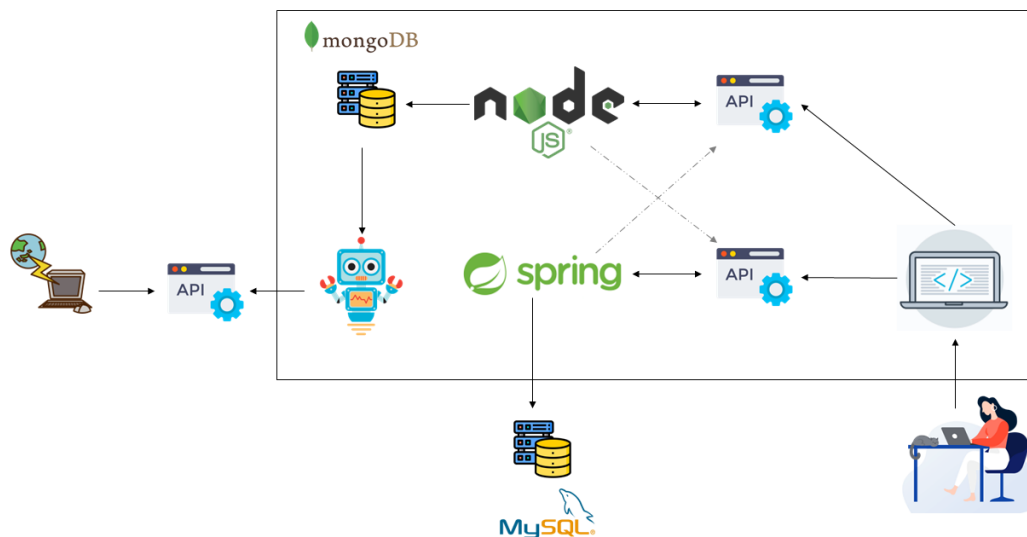


Figure 1. Architecture of the microservices of the project

1.2 Front-End

The scripting language JavaScript is a core technology in web development, along with HTML and CSS. And its surge to the top of the ranks of the world's most prominent programming languages is tied to the rise of the World Wide Web itself. With the popularity of JavaScript comes the increase in the number of front-end frameworks, the most common of which are Angular, Ember.js, Preact, React, Svelte, and Vue.js. Based on results from Stack Overflow's survey, the most broadly used of these frameworks include React, Angular, and Vue.js. Since Angular framework has been tackled during the Isika course, Raiddon will be developed user the latter framework.

1.3 Back-End

Raiddon will include multiple microservices:

- A NodeJS microservice able to exploit data present in a MongoDB database.
- A Spring boot application able to exploit data present in a SQL database.

- A React framework.

2. Objectives of the project

World of Warcraft is a massively multiplayer online role-playing game released in 2004 by Blizzard Entertainment. In world of warcraft, a raid is an organized event in order to down bosses of a specific dungeon. Raids are formed of up to 40 players, so they are pretty hard to put in place. The main objective of a raid is to show off a guild progress of the ongoing game phase and also to gather items dropped from each boss in order to strengthen the players statistics.

Raidon is a website aiming to provide helpful features for world of warcraft classic players. It mainly provides an intuitive and technical solution that connects users who are looking to recruit for or join a guild or a raid. The main objective is to offer services that many players may find useful and that do not exist within the default game interface.

3. Raidon Users

All services offered by Raidon vary based on the status of the player using the website. A user may be registered as:

- A guild master or officer recruiting new members in his guild or looking for players to complete his raid roster on a given date and time.
- A guild member looking for a raid or a new guild.
- A player without a guild looking for a guild or a raid to join.

4. Raidon Services

After adding at least one playable character, a Raidon user may manage their account in the following ways:

- Define availability to join a guild,
- Define availability to participate to a raid,
- Delete Account,
- Update Account,
- Manage equipment,
- Manage professions,
- Manage reputations,
- Add new playable character,
- Manage playable characters,
- Search guild list,
- Contact guild masters,
- Apply to join a guild,
- Evaluate a guild,
- Search available raid list,
- Apply to join a raid,
- Evaluate a raid,
- Search players list,
- Check website blog,

- Check Website tutorials,
- Check attended raids history,
- Create a Guild,
- Organize a Raid Manage Guild,
- Delete player,
- Upgrade player's rank,
- Downgrade player's rank,
- Check players available for raid list,
- Check players looking for guild list,
- Recruit new player,
- Check guild members list,
- Check guild raid history.

5. User issues

- **Guild masters:**

Running a guild is an undertaking and a personal investment, especially when you're just starting out. A guild master has to be active and around as often as possible for their guild to thrive. Being a guild master requires also organising raids at least once a week depending on their guild type. Many options can be found in-game (i.e., the guild interface) , however they remain very limited due to their restriction to connecting to the internet.

- **Players without guild:**

Not having a guild in world of warcraft is indeed challenging when it comes to find a raid. Places in raids are often very hard to get and sometimes impossible.

6. Raiddon Logo



Figure 2. Raiddon Logo

Chapter 2.

Naming Conventions



7. Naming Conventions:

- a. **World of Warcraft (WoW):** World of Warcraft (WoW) is a massively multiplayer online role-playing game (MMORPG) released in 2004 by Blizzard Entertainment.
- b. **Guild:** In video games, a clan, community, guild or faction is an organized group of video game players that regularly play together in one or more multiplayer games.
- c. **Guild master or leader:** A guild leader is a player character who is the head of a guild. They have administrative control of the guild's operations (via "Guild Control" button on the Guild list window), including giving ranks, privileges, adding/removing guild members, etc.
- d. **Guild officer:** An Officer is a rank of member within a Guild that usually has similar privileges to the guild leader. There can be multiple ranks of officers in a guild with varying levels of privilege controlled solely by the guild's leader.
- e. **Guild Tabard:** A tabard refers to a type of clothing bearing a symbol or design and worn in the tabard equipment slot over a character's chest armor. Originally a purely cosmetic item intended to add individuality to a character's look, similar to a shirt, they now often signal an adventurer is championing a cause and thus offer reputation for wearing one in a dungeon.
- f. **Playable characters:** Players control a character avatar within a game world in third- or first-person view, exploring the landscape, fighting various monsters, completing quests, and interacting with non-player characters (NPCs) or other players.
- g. **Faction:** To create a new character, players must choose between the opposing factions of the Alliance or the Horde. Characters from the opposing factions can perform rudimentary communication (most often just "emotes"), but only members of the same faction can speak, mail, group and join guilds. The player selects the new character's race, such as orcs or trolls for the Horde, or humans or dwarves for the Alliance. Players must select the class for the character, with choices such as mages, warriors, and priests available. Most classes are limited to particular races.
- h. **Race:** Race is a term used in World of Warcraft to split various unique creatures into separate groups. Numerous races populate Azeroth including elves, trolls, human, orcs, gnomes, and murlocs, among many others. Many races can interbreed including humans, high elves, night elves, ogres, orcs, and draenei — producing offspring often called half-breeds.
- i. **Class:** A class is the primary adventuring style of a player character. A character's class determines the abilities, powers, skills, and spells they will gain throughout their adventures, and consequently the styles of play available to the character. It determines the types of weapons and armor they can use, which attributes they will value (and how those attributes function), as well as what combat roles the character is suitable for. Class also reflects a

significant choice of path for a character: whether they have chosen to pursue the dark arts of the warlock or the Holy Light of the paladin; the bloody honor of the warrior or the arcane knowledge of the mage.

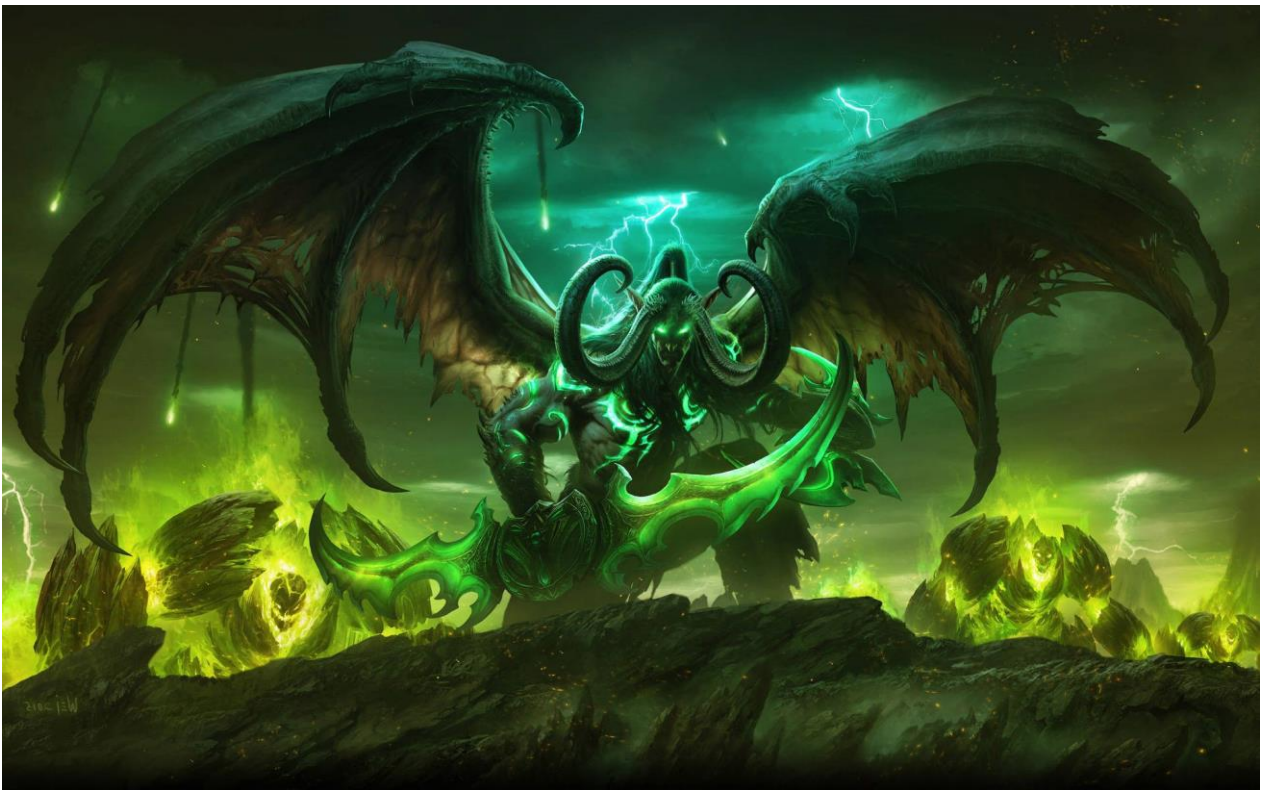
- j. **Profession:** A profession is a trade-oriented set of skills that player characters may learn and incrementally advance in order to gather, make, or enhance items that can be used in World of Warcraft gameplay. In essence, professions are 'jobs' characters may have. Professions are learned and improved via a trainer for a nominal fee, or sometimes advanced with special recipes. Any profession can be learned regardless of a character's faction, race, or class, although some racial traits provide bonuses to a particular profession.
- k. **Reputation:** You can gain or lose favour, otherwise known as reputation, with many of the several different factions in Azeroth. Higher reputation gives access to special rewards or new quests to accomplish.
- l. **Items:** An item is something that a World of Warcraft player character can carry, either in their inventory, represented by an inventory icon, or tracked on a page in the character sheet.
- m. **Raid:** Raids groups are a way to have parties of more than 5 and up to 40 people, divided into up to 8 groups of up to 5 players. The terms "raid" and "raiding" primarily and traditionally refer to PvE raid-specific instances. Raid instances require playing as a team, and are designed to be the most challenging and entertaining PvE content available in the game.
- n. **Loot:** Stuff (treasure: items or money) you get from mobs or containers (barrels, boxes, chests, etc).
- o. **Equipment:** Equipment refers to any item that can be equipped in an equipment slot, as distinguished from items that can only be carried in inventory.
- p. **Loot system:** A Loot System is any method to distribute items amongst a group of players. Whenever two or more players share some reward, they participate in a loot system, even if it's only the default roll system from Blizzard. Frequently, the terms DKP and "loot system" are used interchangeably, because DKP based systems are the most widely used type of formal loot systems.
- q. **Boss:** Boss is a quite general phrase used for several special types of mobs. A common characteristic for all bosses is that they have a unique name and appear only once in the game. Bosses are harder to kill than the "normal" elite mob of the same level, and almost all bosses are immune to Crowd Control, though some may be susceptible in accordance with their intended mechanics

-
- r. **Mobs:** A mob (short for mobile) is a generic term for any non-player entity whose primary purpose is to be killed for experience, quest objective, or loot.
 - s. **Instance:** Major hostile areas like dungeons, keeps, and other confined areas can have sub-areas called instances (aka instance(d) dungeons). These instances are special areas in the World of Warcraft where your group or raid party is able to interact with a dungeon privately; that is, without interference from other parties or raids.
 - t. **Quest:** A quest is a task given to a player character that yields a reward when completed. Most quests are given by an NPC (non-player character).

Chapter 3.

Functional Requirements

- Business process model
- Use case
- Class diagram



8. Business Process Model

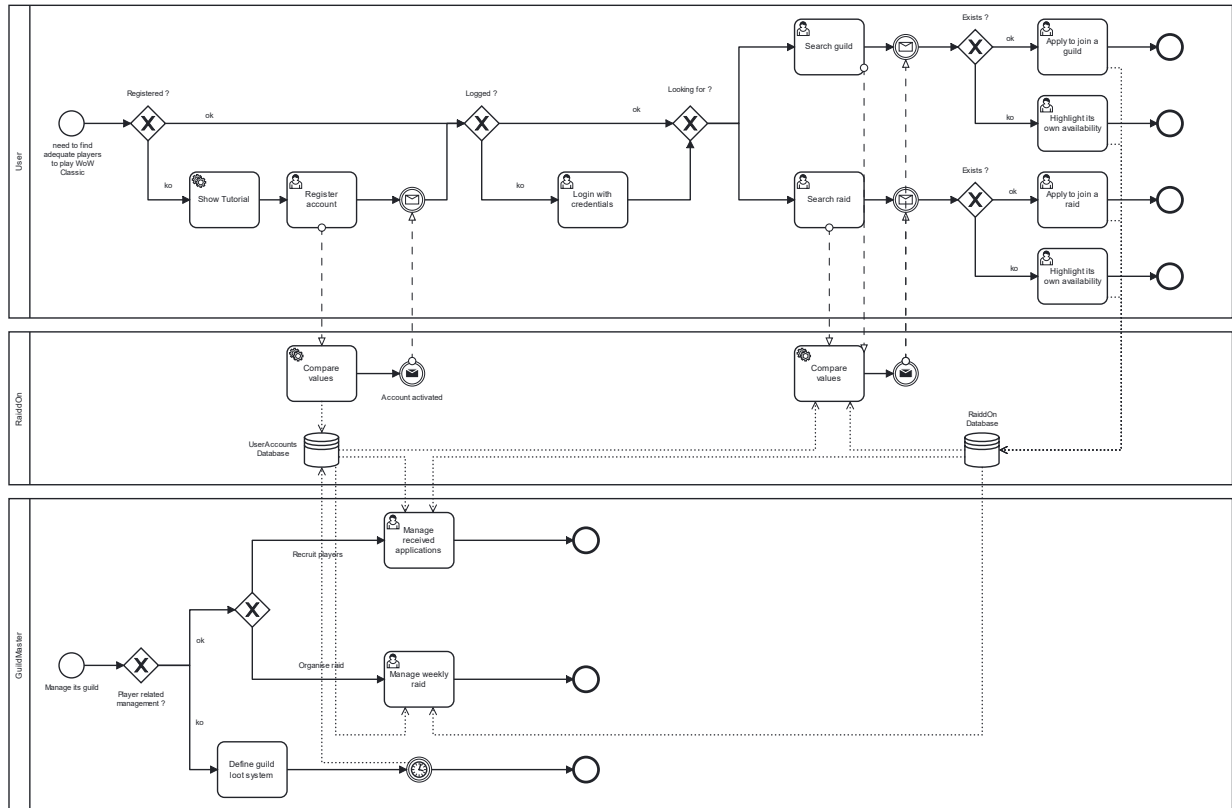


Figure 3. Business Process Model for Raiddon

The business process model (figure 3) represents the sequence of activities to be carried out by Raiddon users. The beginning of the for mentioned sequence of activities is marked by the need of a World of Warcraft (hereafter, WoW) player to find other players sharing the same in-game objectives. Each player must have a Raiddon account and at least one playable character added to their account.

A logged-in user has access to a detailed guilds page in which they can search all registered guild on Raiddon. This ultimately allows the user to apply to join one of the guilds that satisfies his in-game ambitions. Alternatively, the same user can also search raids and therefore apply to join a raid if they are eligible to do so. It is to note that each application require approval from the guild master and the raid organizer respectively.

If the user is not admitted in the guild nor the raid or they do not find any results that match their needs they can do one of the following actions by updating their profile:

- Define their availability to raid,
- Define their availability to join a guild.

Another sequence of activities is marked by the need of a WoW guild master player to manage their guild which is challenged by two factors: one related to managing the guild's members (i.e., delete, downgrade or upgrade player) and one related to managing the guild's core value (i.e., guild loot system).

A wow guild master is also faced by the obligation to provide weekly raids for their members. To do so they have to organise raids as well as manage received applications from players wishing to join the guild.

9. Use Cases (UC)

9.1 User account management

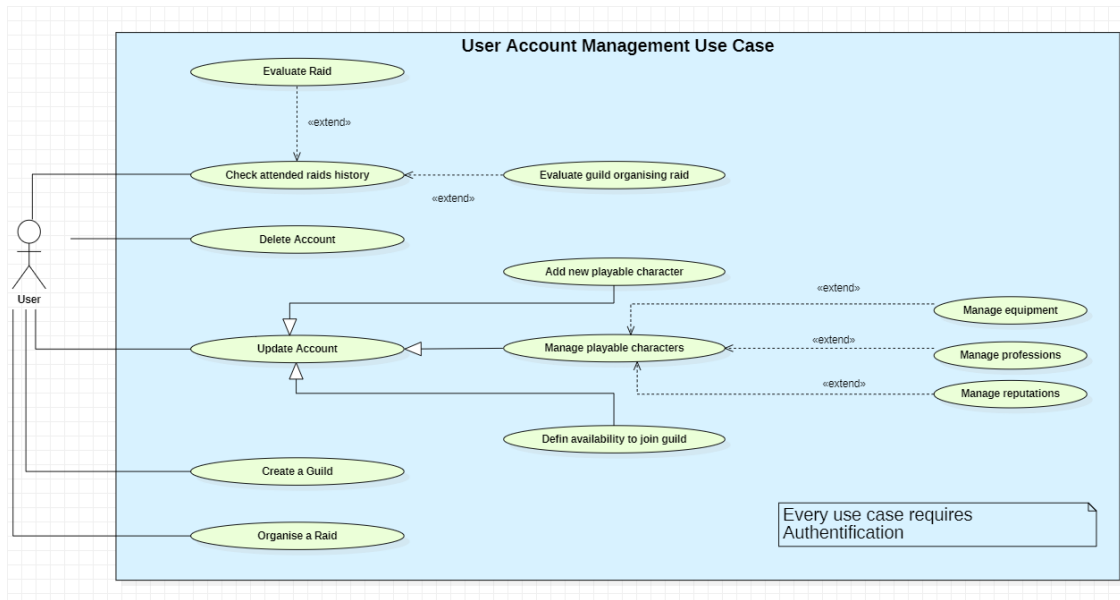


Figure 4. Overview of all use cases that can be carried out by a user who is not a guild master or officer.

9.1.1 Create Account

Use case name	Create Account
ID	CU-1
Objective	Create an account on Raiddon website
Main Actors	Guest user
Secondary actors	N / A
Description	CU-1 begins when the main actor wishes to create an account. They create their username, password and fill in the requested information. CU-1 ends when the mandatory information is validated and registered in the management system
Pre-conditions	The main actor must not already have an account and the interface must be accessible.
Post-conditions	1. The main actor changes status and now becomes a Raiddon user. 2. The user's account is registered in the organization's management system.
Scenario	1. The main actor opens the signup page 2. The system displays the fields to be completed under a registration form: <ul style="list-style-type: none"> • Username • Password

	<ul style="list-style-type: none"> • Password confirmation • E-mail address • Checkbox in order to accept terms & conditions • The guest fills in and submit the form. <ol style="list-style-type: none"> 3. The system generates a standard profile picture for the profile 4. The system displays the home page of the logged in member
Exceptions	The account already exists.
Constraints	<ol style="list-style-type: none"> 1. CS1: The main actor must accept the terms of use. 2. CS2: All fields are mandatory.
Requirements	N / A

9.1.2 Sign-in to account

Use case name	Sign-in to account
ID	CU-2
Objective	Sign in to Raiddon account
Main Actors	Raiddon registered user
Secondary actors	Raiddon management system
Description	CU2 begins when a Raiddon user wishes to access to their account and it ends when the Raiddon management system displays the account
Pre-conditions	The main actor must have an account
Post-conditions	<ol style="list-style-type: none"> 1. The main actor has access to their account and its functionalities again 2. In scenario 3 the management system is updated with the new password.
Scenarios	
Scenario 1: Nominal	<ol style="list-style-type: none"> 1. The main actor goes to their Profile page. 2. The main actor choses the login/sign-in option. 3. The system displays the required fields to fill in: <ul style="list-style-type: none"> • Username • Password 4. The main actor fills in the fields. 5. Credentials are recognised by Raiddon. 6. The system displays the actor's account
Scenario 2: Credentials are not recognised	<ol style="list-style-type: none"> 1. The main actor goes to the website homepage. 2. The main actor choses the login/sign-in option. 3. The system displays the required fields to fill in: <ul style="list-style-type: none"> • Username • Password 4. The main actor fills in the fields. 5. Credentials are not recognised by Raiddon.

	6. The system displays an error message. 7. All fields are reset.
Scenario 3: The actor forgot their password	1. The main actor goes to the website homepage. 2. The main actor chooses the login/sign-in option. 3. The system displays the required fields to fill in: <ul style="list-style-type: none"> • Username • Password 4. The main actor chooses the option "I forgot my password". 5. The system displays the fields to fill in: User account email address. 6. The main actor fills in the field and validates. 7. The main actor receives a new temporary password by e-mail. 8. The main actor logs in with their new password. 9. The system displays the actor's account 10. The system informs the actor that they must change their temporary password shortly.
Exceptions	N / A
Constraints	N / A
Requirements	N / A
Included UC	UC-1

9.1.3 Manage account

Use case name	Manage account
ID	CU-3
Objective	1. Define availability to join a guild 2. Define availability to participate to a raid 3. Delete Account 4. Update Account 5. Manage equipment 6. Manage professions 7. Manage reputations 8. Add new playable character 9. Manage playable characters
Main Actors	Raiddon registered user
Secondary actors	Raiddon management system
Description	CU3 begins when a Raiddon user wishes to modify the information contained in their profile. The CU ends following the modifications.
Pre-conditions	The main actor must have an account and must be connected
Post-conditions	The main actor's personal information is updated.
Scenarios	

Scenario 1: Nominal	<ol style="list-style-type: none"> 1. The main actor goes to their Profile page. 2. The main actor selects “Edit my profile”. 3. The system displays: <ul style="list-style-type: none"> • Different pre-filled fields: <ul style="list-style-type: none"> ○ Username ○ E-mail address • Playable character section: <ul style="list-style-type: none"> ○ Character username ○ Character race ○ Character class ○ Character faction ○ Character specialisation ○ Character Professions • A link allowing to upload a new profile picture. 4. The main actor edits one or more fields. 5. The main actor validates the modifications. 6. The system displays the modified profile.
Exceptions	<ol style="list-style-type: none"> 1. Ex1: If the information is syntactically invalid, the member must correct the fields in error. 2. Ex2: If the user has not filled in all the necessary fields, they must enter the information requested by the system.
Constraints	<ol style="list-style-type: none"> 1. CS3: All fields must be correctly filled in. 2. CS4: The profile image must be at least 200x200px, at most 400x400px, PNG/GIF/JPEG format and maximum size 2MB.
Requirements	N / A
Included UC	UC-2

9.1.4 Deleting account

Use case name	Deleting account
ID	CU-4
Objective	Deleting all information on Raiddon account
Main Actors	Raiddon registered user
Secondary actors	Raiddon management system
Description	CU4 begins when a Raiddon user wishes to close their account and delete all their information from the website. To do this, they access this functionality from their profile. The CU ends when the user has finished his deletion request.
Pre-conditions	The main actor must have an account and must be connected
Post-conditions	The main actor's account is deleted

Scenarios	
Scenario 1: Nominal	<ol style="list-style-type: none"> 1. The main actor goes to their profile and validates the “delete my account” option. 2. The system displays: <ul style="list-style-type: none"> • «Are you sure you want to delete your account? This action is irreversible, data such as raid history will no longer be available” • A link “yes, I really want to delete my account” to validate the operation • A link "no, I want to keep my account" invalidating and returning to the profile. 3. The main actor checks the appropriate box.
Exceptions	N / A
Constraints	CS5: The copy of the account present in the archive changes status and becomes inactive
Requirements	N/A
Included UC	UC-2

9.1.5 Search available raid

Use case name	Search available raid
ID	CU-5
Objective	Search all available raid lists in order to join one
Main Actors	Raidon registered user
Secondary actors	Raidon user organising the raid
Description	CU5 begins when a Raidon user wishes to join a raid using the search raid tool. The CU ends when the user applies for a raid and waits the organising user response.
Pre-conditions	<ol style="list-style-type: none"> 1. The main actor must have an account and must be connected 2. The main actor must have at least added one playable character to their account.
Post-conditions	A “join raid” request is sent from the main actor of the CU to the secondary actor of the CU.
Scenarios	
Scenario 1: Nominal	<ol style="list-style-type: none"> 1. The main actor consults the available raids list 2. The main actor choses a raid that they wish to join 3. The system verifies that the actor satisfies all the required conditions in order to join the raid in question. 4. The system displays to the actor all the playable characters with which they are eligible to join the raid. 5. The main actor choses the playable character they wish to raid with. 6. The main actor now has a pending “join raid” application in their profile.

	7. The raid organiser is notified and the “join raid” application is added to their “manage join raid applications” section.
Exceptions	N/A
Constraints	CS5: The copy of the account present in the archive changes status and becomes inactive
Requirements	N / A
Included UC	UC-2, UC-3

9.1.6 Check attended raids history

Use case name	Check attended raids history
ID	CU-6
Objective	Search all available raid lists in order to join one
Main Actors	Raidon registered user
Secondary actors	N / A
Description	CU6 begins when a Raidon user wishes to check their past raids history.
Pre-conditions	3. The actor must have an account and must be connected 4. The actor must have found at least one raid via Raidon
Post-conditions	N / A
Scenarios	
Scenario 1: Nominal	<ol style="list-style-type: none"> 1. The main actor goes to the page corresponding to his raid history from their profile. 2. The system displays for all linked raids: <ul style="list-style-type: none"> • Raid id • Raid date • Raid time • Raid name • Raid encounters • A link towards the warcraft logs page.
Exceptions	N / A
Constraints	N / A
Requirements	N / A
Included UC	UC-2, UC-3

9.1.7 Organise a raid

Use case name	Organise a raid
ID	CU-7
Objective	The main actor wishes to organise a raid. They can define the raid name, date, time and duration.
Main Actors	Raidon registered user

Secondary actors	N / A
Description	CU7 begins when a Raiddon user wishes to organise a raid or an instance. It ends when the raid is published on the website
Pre-conditions	<ol style="list-style-type: none"> 1. The actor must have an account and must be connected 2. The actor must have found at least one playable character in their account.
Post-conditions	The raid is published on the website and added to the available raids list.
Scenarios	
Scenario 1: Nominal	<ol style="list-style-type: none"> 1. The main actor follows the steps of CU-2. 2. The main actor choses the organise raid option. 3. The system verifies that the main actor satisfies all the requirements in order to organise a raid. 4. The system displays the list of fields to fill in: <ul style="list-style-type: none"> • Raid name • Raid date and time • Raid duration • Raid loot system • Raid requirements • Comments 5. The main actor fills in the fields and submit the form. 6. The system displays: "Your raid has been published on the website"
Scenario 1: The system does not validate the raid creation	<ol style="list-style-type: none"> 1. The main actor follows the steps of CU-2. 2. The main actor choses the organise raid option. 3. The system verifies that the main actor satisfies all the requirements in order to organise a raid. 4. The system displays the error message: "Please add a playable character to your account in order to organise a raid"
Exceptions	EX3: If the information entered is incorrect, the system displays an error message.
Constraints	N / A
Requirements	N / A
Included UC	UC-2, UC-3

9.1.8 Create a guild

Use case name	Create a guild
ID	CU-8
Objective	The main actor wishes to create a guild. They can define the guild name, type, loot system and emblem.
Main Actors	Raiddon registered user

Secondary actors	N / A
Description	CU8 begins when a Raiddon user wishes to create a guild. It ends when the guild is added to the active guilds list on the website.
Pre-conditions	The actor must have an account and must be connected
Post-conditions	The guild is added to the active guilds list on the website.
Scenarios	
Scenario 1: Nominal	<ol style="list-style-type: none"> 1. The main actor follows the steps of CU-2. 2. The main actor choses the create guild option. 3. The system verifies that the main actor satisfies all the requirements in order to organise a raid. 4. The system displays the list of fields to fill in: <ul style="list-style-type: none"> • Guild name • Guild gameplay type • Guild raids per week • Guild emblem • Guild loot system • Guild objectives • Guild website 5. The main actor fills in the fields and submit the form. 6. The system displays: "Your request is pending". 7. The system will check if the guild exists and if the main actor is the real guild master. 8. The system notifies the main actor that their guild is created and added to the active guilds list on the website.
Exceptions	EX3: If the information entered is incorrect, the system displays an error message.
Constraints	N / A
Requirements	N / A
Included UC	UC-2, UC-3

9.2 Guild master account management

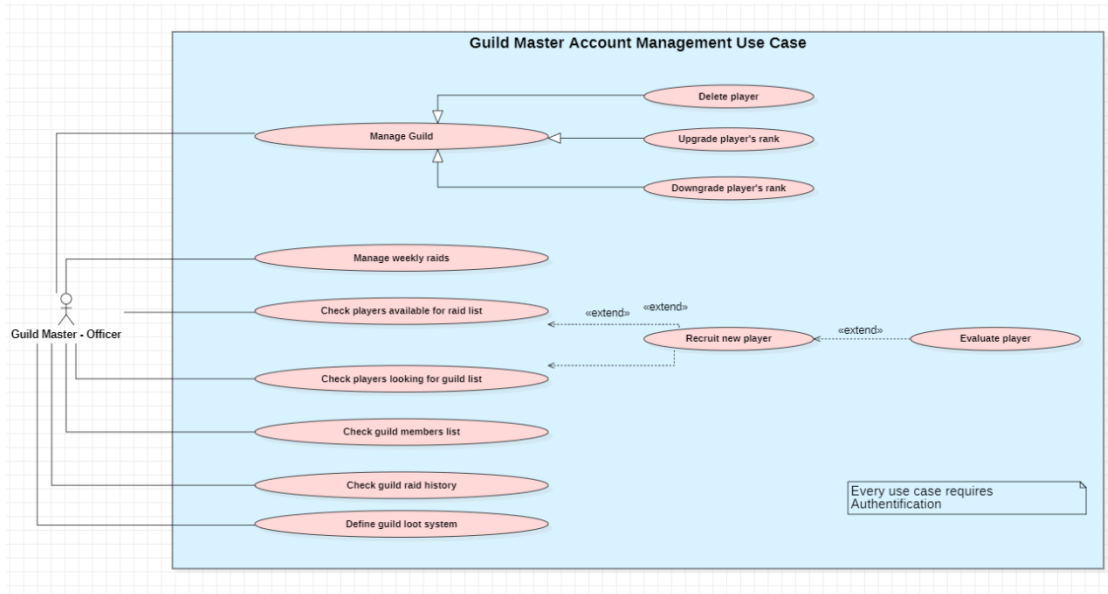


Figure 5. Overview of all use cases that can be carried out by a user who is a guild master or officer.

9.2.1 Manage guild

Use case name	Manage account
ID	CU-9
Objective	<ol style="list-style-type: none"> 1. Upgrade a player's rank 2. Downgrade a player's rank 3. Delete a player from member list
Main Actors	Raiddon registered user
Secondary actors	N / A
Description	CU-9 begins when a Raiddon user wishes to manage their guild members.
Pre-conditions	<ol style="list-style-type: none"> 1. The main actor must have an account and must be connected. 2. The main actor must be a guild master.
Post-conditions	<ol style="list-style-type: none"> 1. The guild information is updated.
Scenarios	
Scenario 1: Nominal	<ol style="list-style-type: none"> 1. The main actor goes to their Profile page. 2. The main actor selects "Manage guild". 3. The system displays the following options: 4. Update guild details 5. Update guild member's list 6. The main actor choses "update guild details" 7. The system displays the following prefilled fields: <ul style="list-style-type: none"> • Guild name • Guild emblem

	<ul style="list-style-type: none"> • Guild schedule • Guild loot system <p>8. The main actor edits one or more fields.</p> <p>9. The main actor validates the modifications.</p>
Scenario 2: The guild master choses to update members list	<p>1. The main actor goes to their Profile page.</p> <p>2. The main actor selects “Manage guild”.</p> <p>3. The system displays the following options:</p> <p>4. Update guild details</p> <p>5. Update guild member’s list</p> <p>6. The main actor choses “update guild member’s list”</p> <p>7. The system displays a list of all members of the guild from which the main actor can select a member.</p> <p>8. The main actor selects the member of their choice.</p> <p>9. The system displays three options:</p> <ul style="list-style-type: none"> • Upgrade a player’s rank • Downgrade a player’s rank • Delete a player from member list <p>10. The main actor choses one of the three options.</p> <p>11. The main actor validates the modifications.</p>
Exceptions	<p>1. Ex1: If the information is syntactically invalid, the member must correct the fields in error.</p> <p>2. Ex2: If the user has not filled in all the necessary fields, they must enter the information requested by the system.</p>
Constraints	<p>1. CS3: All fields must be correctly filled in.</p> <p>2. CS4: The guild emblem must be at least 200x200px, at most 400x400px, PNG/GIF/JPEG format and maximum size 2MB.</p>
Requirements	N / A
Included UC	UC-2, UC-8

9.2.2 Check players available for raid list

Use case name	Check players available for raid list
ID	CU-10
Objective	To be able to consult all Raiddon users with the option “available for raid” turned on.
Main Actors	Raiddon registered user
Secondary actors	N / A
Description	CU-10 starts when the main actor wishes to check players available for raid. It ends when the system displays the list of players available for raid.
Pre-conditions	The main actor must have an account and must be connected.

Post-conditions	N / A
Scenarios	
Scenario 1: Nominal	<ol style="list-style-type: none"> 1. The main actor signs in (CU-2). 2. The main actor navigates to the “Players” page. 3. The main actor choses the option “Players available for raid”. 4. The system displays the list of all players available for raid.
Exceptions	N / A
Constraints	N / A
Requirements	N / A
Included UC	UC-2

9.2.3 Check players looking for guild list

Use case name	Check players looking for guild list
ID	CU-11
Objective	To be able to consult all Raiddon users with the option “looking for guild” turned on
Main Actors	Raiddon registered user
Secondary actors	N / A
Description	CU-11 starts when the main actor wishes to check players looking for a guild. It ends when the system displays the list of players looking for a guild.
Pre-conditions	The main actor must have an account and must be connected.
Post-conditions	N / A
Scenarios	
Scenario 1: Nominal	<ol style="list-style-type: none"> 1. The main actor signs in (CU-2). 2. The main actor navigates to the “Players” page. 3. The main actor choses the option “Players looking for guild”. 4. The system displays the list of all players looking for guild.
Exceptions	N / A
Constraints	N / A
Requirements	N / A
Included UC	UC-2

9.3 Raiddon Overview

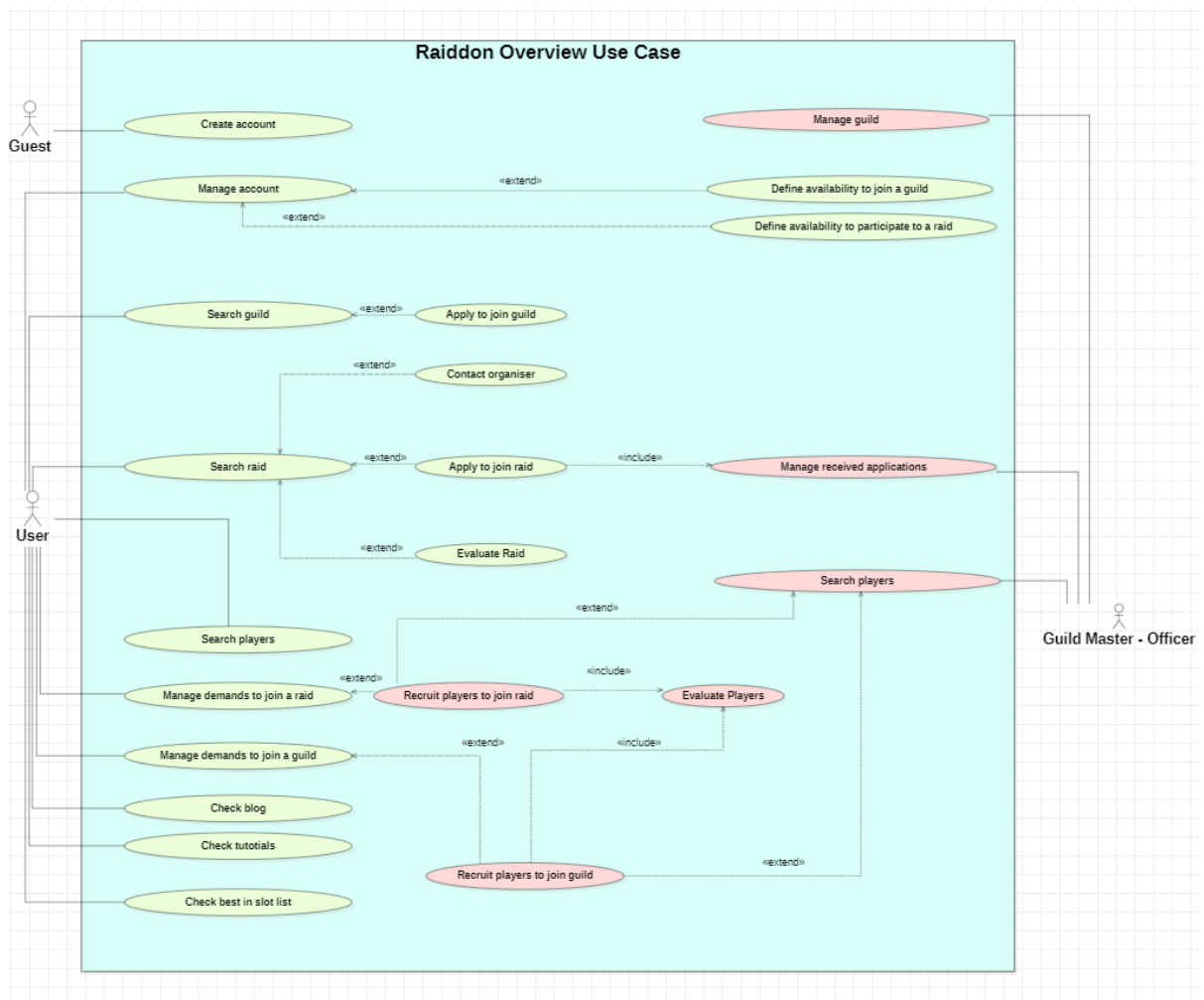


Figure 6. Overview of all use cases that can be carried out using Raiddon. Green UCs represent all actions carried out by a user who is not a guild master while all the red ones represent actions carried out by a guild master or officer.



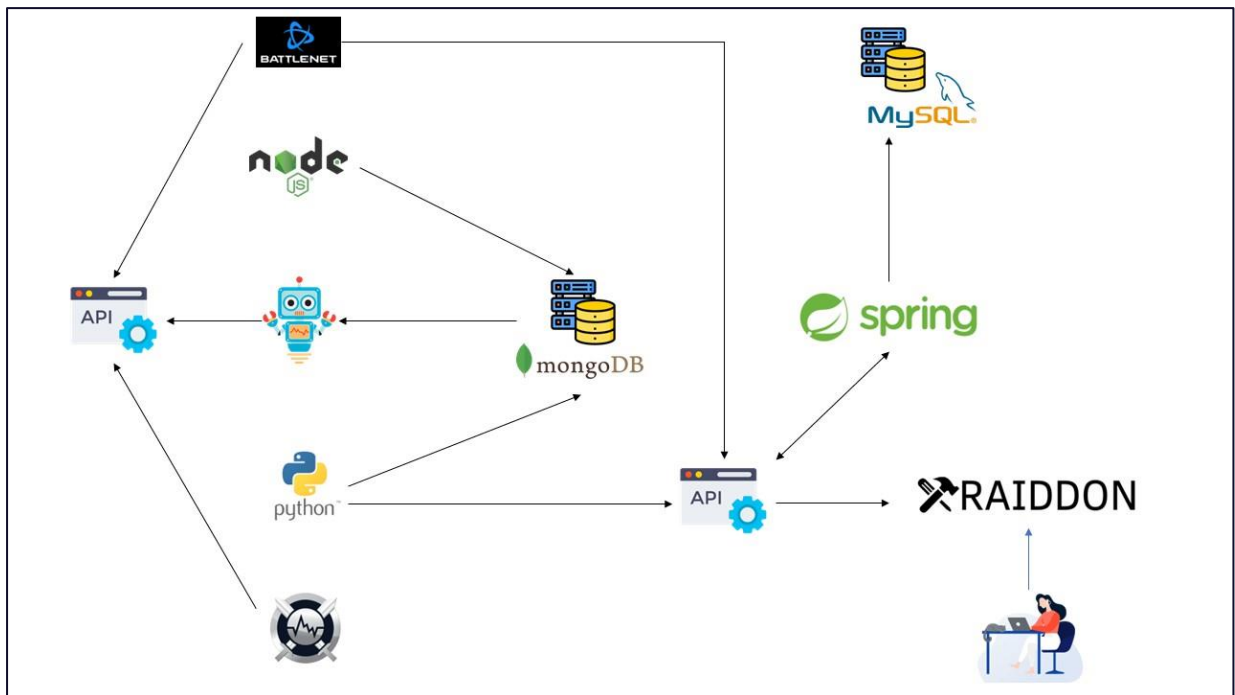
Chapter 4.

Data Fetching

- Raiddon architecture
- Fetching access tokens
- Fetching data from Battlne.net
- Fetching data from Warcraft Logs



10. Raiddon Architecture



11. Fetching access tokens:

Battle.net and Warcraft Logs use OAuth 2.0 for API authentication. OAuth allows clients to request and then use an access token to authenticate API requests. The first step in using OAuth is getting a **client_id** and **client_secret** by creating a new client on both websites.

The OAuth 2.0 Authorization Code Flow allows an application to access a user's data on their behalf. This allows an application to acquire more sensitive, opt-in information about a user, such as a user's private reports, after obtaining the user's permission to do so. The Authorization Code Flow has two major parts: the authorization code request and the access token request.



Figure 8. OAuth 2.0 schema

Access tokens last for 24 hours. A user changing their password, removing the authorization for an application's account, or getting their account locked for any reason, results in the expiration of their current access tokens. Developers should always check the response and request a new access token if the current one fails to work.

12. Fetching data from Battle.net

The World of Warcraft APIs provide developers with tools to enrich any fan site with information about World of Warcraft, StarCraft II, Diablo III, and Hearthstone. They can be integrated into websites in several ways:

- Add World of Warcraft character achievements, guild information, and Mythic Dungeon leaderboards.
- Display player profiles, ladder information, and match history for StarCraft II.
- Retrieve Diablo III player hero information, season leaderboards, and character class and skill information.
- Return information about Hearthstone cards and decks.

The following OAuth endpoints require an access token retrieved via the OAuth authorization code flow:

- GET /oauth/userinfo
- GET /profile/user/wow
- GET /profile/user/wow/protected-character/{realm-id}-{character-id}
- GET /profile/user/wow/collections
- GET /profile/user/wow/collections/pets
- GET /profile/user/wow/collections/mounts

The code used to fetch battle.net data in this project can be found on this [GitHub Repository](#). It is a flexible and feature-rich Javascript that allows fetching the access token using the [BlizzAPI](#). It also allows access to all Battle.net data-base.

13. Fetching data from warcraft logs

OAuth requires two commonly-used URIs to work properly. The `authorize_uri` gets user authorization to allow applications to access certain information, such as the fights and events of a private report. The `token_uri` submits items, such as authorization codes, to request access tokens applications can use with the v2 API.

- The Authorization URI is: <https://www.warcraftlogs.com/oauth/authorize>
- The Token URI is: <https://www.warcraftlogs.com/oauth/token>

The v2 APIs require access tokens granted one of three possible flows: the client credentials flow or the authorization code flow, or the PKCE code flow.

The code used to fetch battle.net data in this project can be found on this [GitHub Repository](#).

Chapter 5.

Spring Boot Server



Chapter 6.

Angular Framework

