System Specification

Project Name	Discord Bot
Project Manager	Philip Trinkl
Document Owner	Fabian Tischler
Created on	27/11/2018 8:14
Last modified	20/12/2018 14:12
State	X in Progress Released

Content

1. Initial Situation and Goal

1.1 Initial Situation

Right now, there are many bots with only a few functions. Our idea is to make a big bot with many functions to make it easier for the user. It is always a bit annoying that there are many bots with different prefixes and many functions are on the server more than once due to the big number of bots. With our bot we try to take most (the most important) functions from the well-known bots and put them together on our bot.

1.1.1 Application Domain

Our application domain is Discord. Bots are designed and programmed for users who want to simplify their use of Discord. They are used for special functions which are not featured by Discord itself, such as playing music or caring a pet. The bots are used on Discord servers, which every user can create - there is no limit of different servers or bots. To use a function the command has to be written in the text channels and if it is valid the command will be performed. Beside that there are also voice channels where you can talk to other users but it is not possible to start a command with your voice.

1.1.2 Glossary

<Prefix> <With this you execute the bot commands. Examples: Exclamation Mark, Point, Plus, …>

<Discord Bot> <A Discord bot is a tool which makes the use of a Discord server easier>

<Discord> <A voice chat especially for gamers>

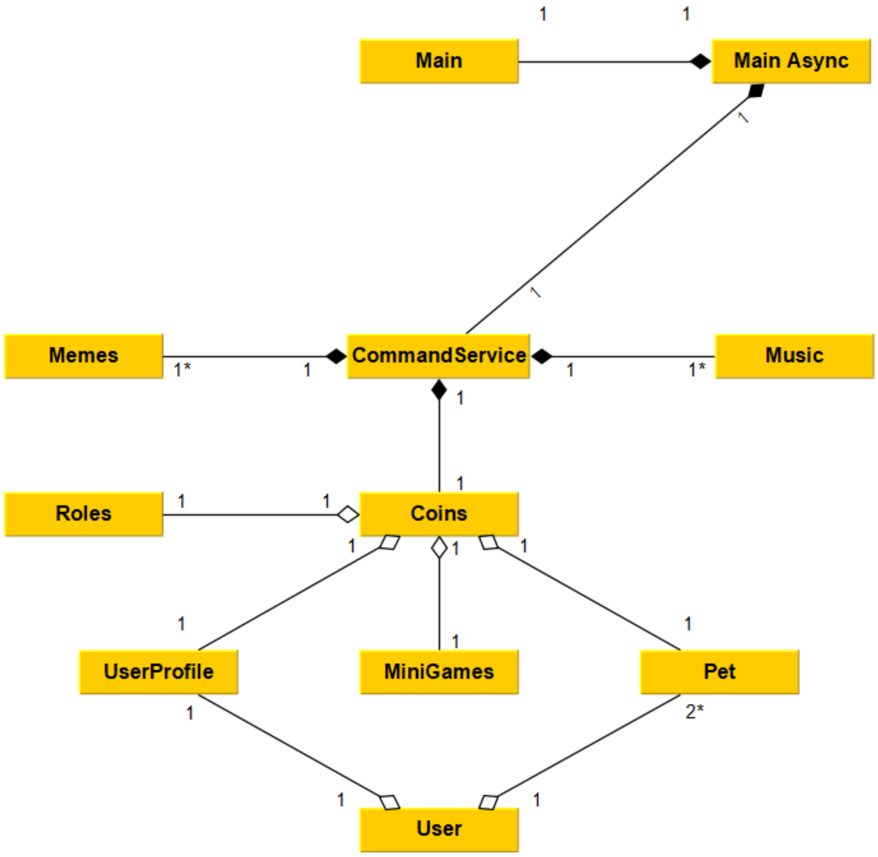
<Discord Server> <It’s free and every user can create one. Users can invite other users to join their server and admins of the server can add Discord bots>

<Channel> <Can be either a text based channel or a voice based channel>

<text channel> <It’s like a chat program, everyone who joined the server can send and read messages>

<voice channel> <You can talk to other users>

1.1.4 Model of the Application Domain

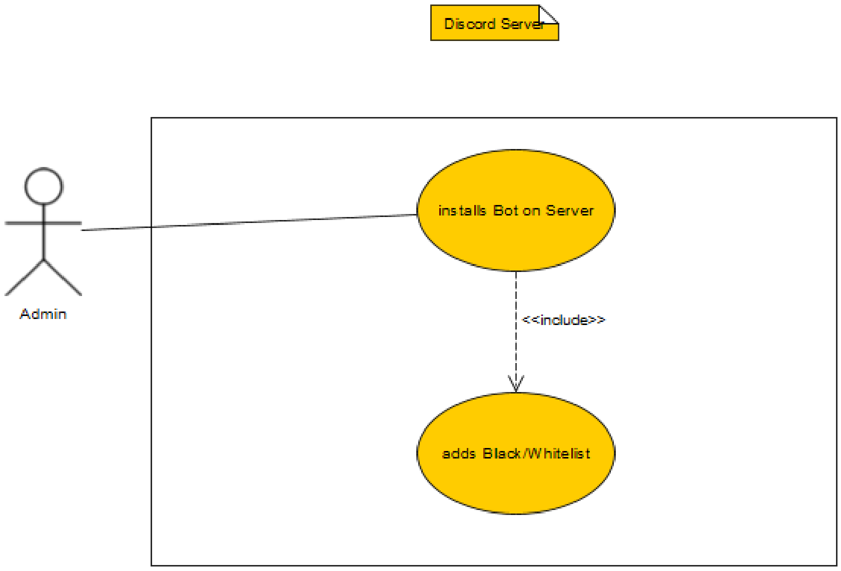


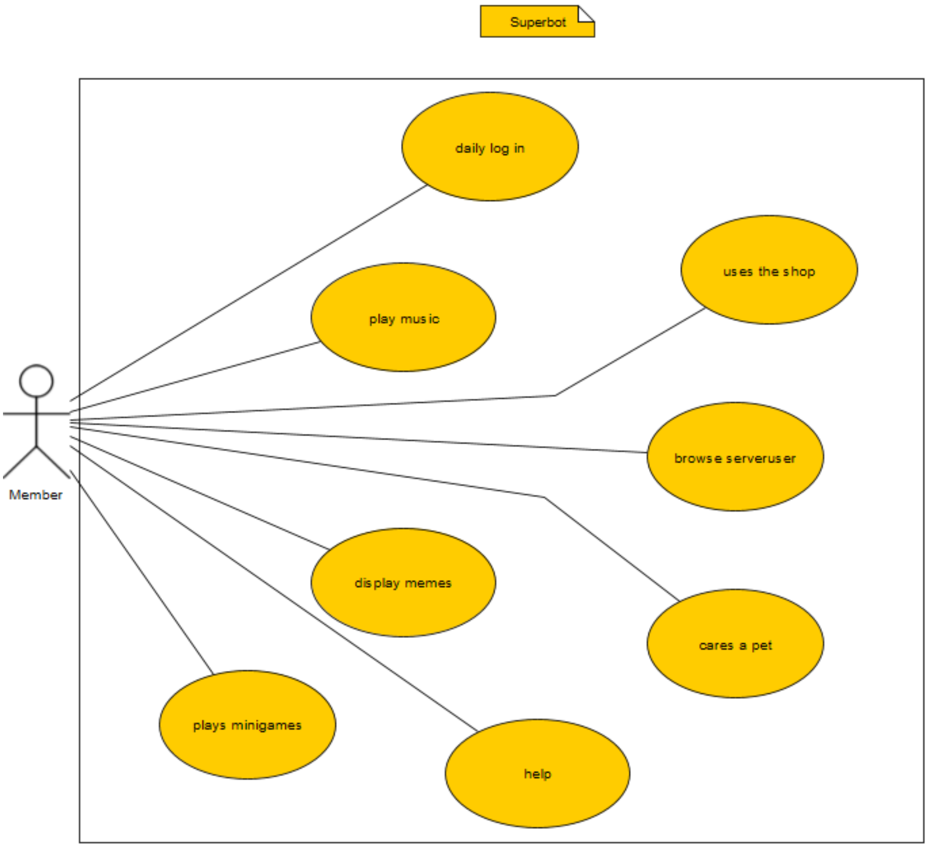
1.2 Goal Definition

The main goal of the bot is to unify and summarise the most wanted commands for discord users. The problem of the bots on Discord is that every bot is specialized on different command types. For example the Discord bot „Music“ is specialized on playing music. The user wants to use this bot but also the bot „Games“ which is specialized on playing games and has again different prefixes. So in order to use the functions of the two bots the user has to know the prefixes of the functions with which one can execute them and has to switch between those two bots. Our bot will be developed for removing this problem and creating a prefix with which the user can execute every command on our bot.

2. Functional Requirements

2.1 Use Case Diagrams





2.2 Details

<Discord Server-Use Case Details>

This use case describes how the admin gets the bot on the server. First, he installs the bot on his server and then he adds a white- and a blacklist of who can use the bot.

<Superbot-Use Case Details>

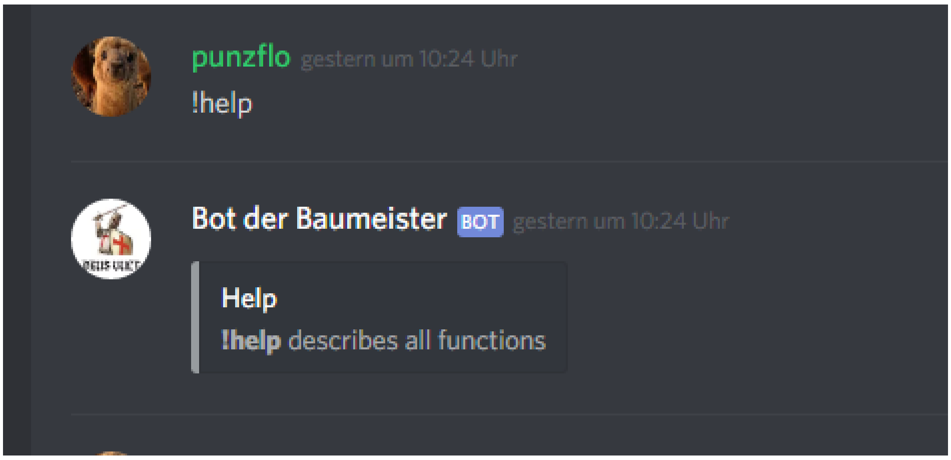
This use case describes what the different functions of the super bot are. Play Music is the function which allows the user to play music for example from YouTube or Soundcloud. Use the shop is a function where the user can spend coins which can be collected from daily log in or games and there one can get pets or a role. With browse serveruser one can display a specific user from the server. Display memes is a function which is used to display memes from a database in a text channel. With the minigame command a user can play a few different minigames and collect coins. The help function is there to explain the different functions to the user. The daily log in function has to be used every day in order to earn coins. The cares a pet function is used when one wants to feed, pat, play, wash or care otherwise for the pet.

2.2.1 Characteristic Information

Discord Server Use Case <ID 001>	
Goal:	The goal is to show how the admin gets the bot on the server
Precondition:	The bot is not on the server
Postcondition:	The bot is on the server
Involved User:	Admin, on some servers also members
Triggering Event:	Installing the server

Superbot Use Case <ID 002>	
Goal:	The goal is to show the different functions
Precondition:	The user wants to use a function
Postcondition:	The user used a function
Involved User:	Member/User
Triggering Event:	Command

2.2.2 GUI to call the use case



punzflo gestern um 10:24 Uhr

!help

Bot der Baumeister BOT gestern um 10:24 Uhr

Help

!help describes all functions

Input field	Valid inputs
Text bar	Has to start with the right prefix, command must exist, parameters have to be valid too (validity check is different at every command)

2.2.3 Scenario for the standard use (good case)

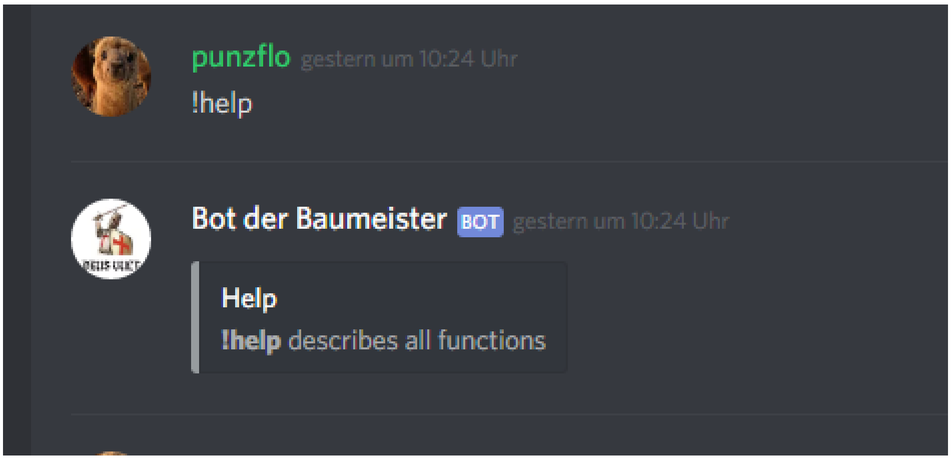
<Use Case 1, Discord Server>

Step	User	Activity
<Step-1.>	<Admin>	<invite the bot to the server>
<Step-2.>	<Admin>	<sets up the white/blacklist>

<Use Case 2, Superbot>

Step	User	Activity
<Step-1.>	<Member of the Server>	<writes the command in the text box>
<Step-2.>	<Member of the Server>	<gets output>

2.2.4 GUIs for the standard use



punzflo gestern um 10:24 Uhr

!help

Bot der Baumeister BOT gestern um 10:24 Uhr

Help

!help describes all functions

Input Field	Valid Inputs
Text bar	Has to start with the right prefix, command must exist, parameters have to be valid too (validity check is different at every command)

2.2.5 Scenarios for non-standard uses (bad cases or work around cases)

<Error Case 1>

Step	User	Activity
<Step 1>	<Member of a server >	<types in wrong command or wrong prefix>
<Step 2>	<Member of a server >	<gets error message>

<Error Case 2>

Step	User	Activity
<Step 1>	<Member of a server >	<types in a YouTube URL which is locked in the country where the bot is hosted>
<Step 2>	<Member of a server>	<gets error message>

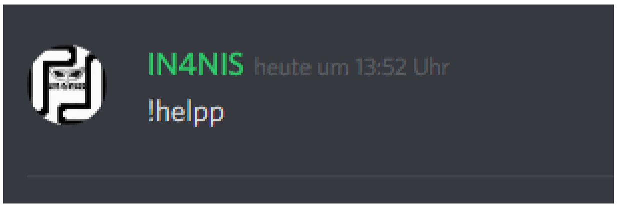
<Error Case 3>

Step	User	Activity
<Step 1>	<Admin >	<invites bot on the server >
<Step 2>	<Admin >	<wants to add Black/Whitelist but the Bot is currently offline>

<Error Case 4>

Step	User	Activity
<Step 1>	<Member of a server >	<types in any command but the bot is currently offline >

2.2.6 GUIs for the non-standard uses

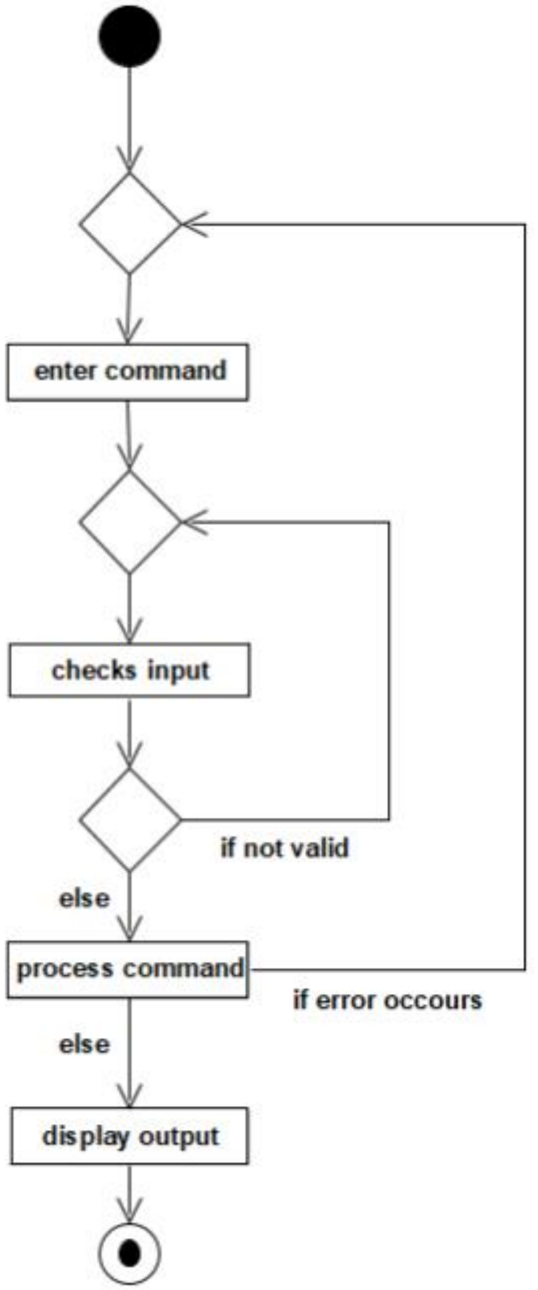


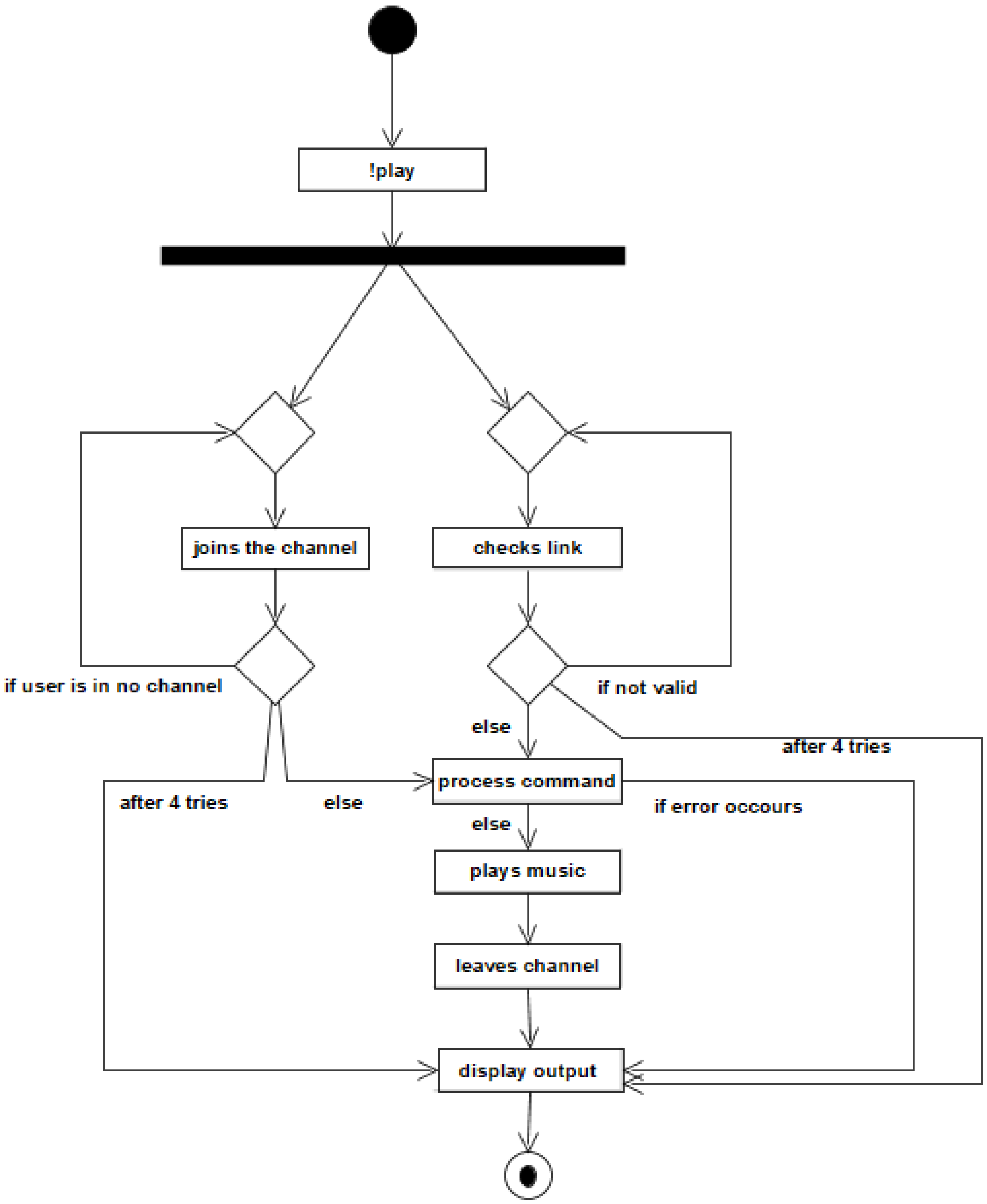
IN4NIS heute um 13:52 Uhr

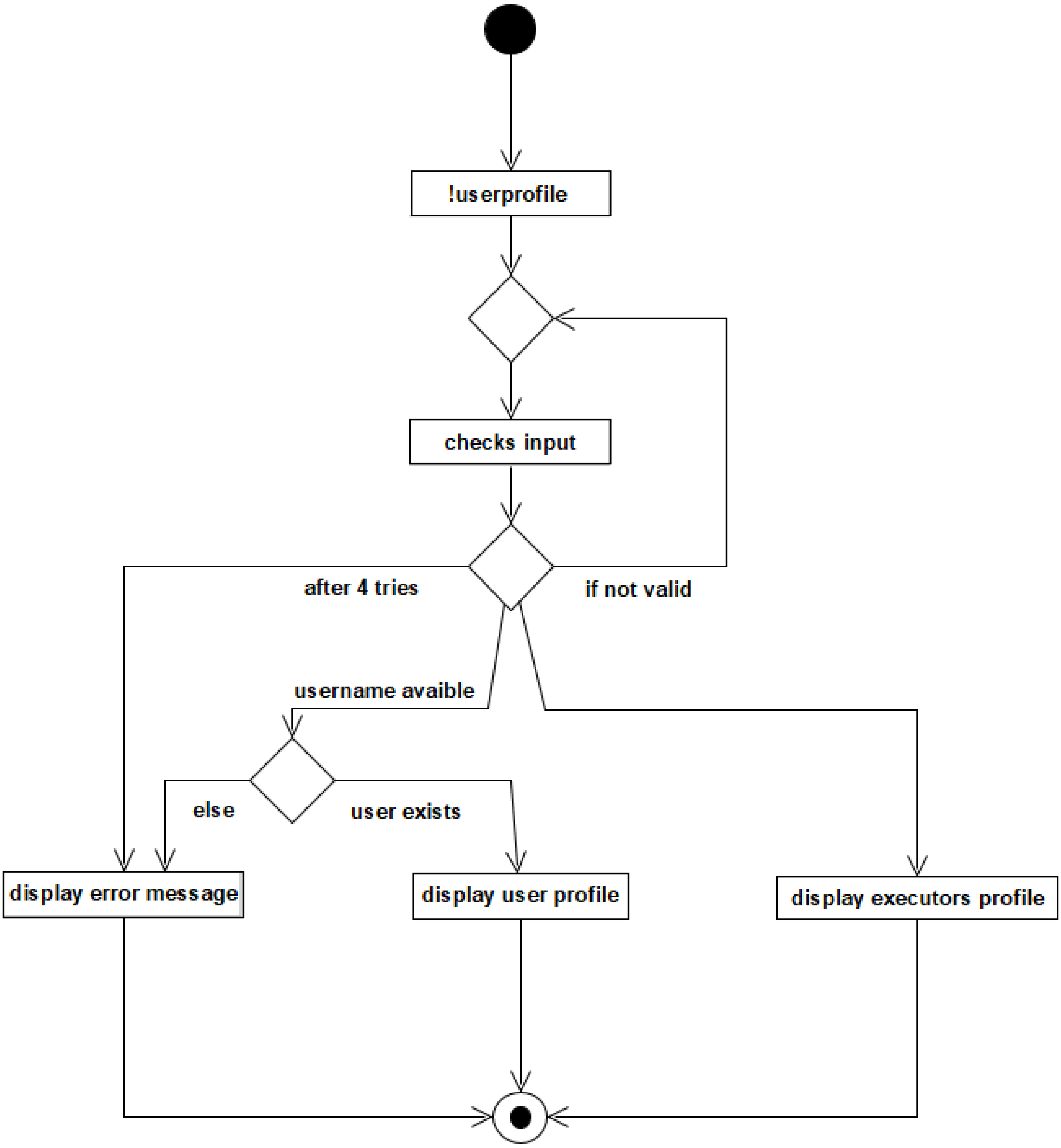
!helpp

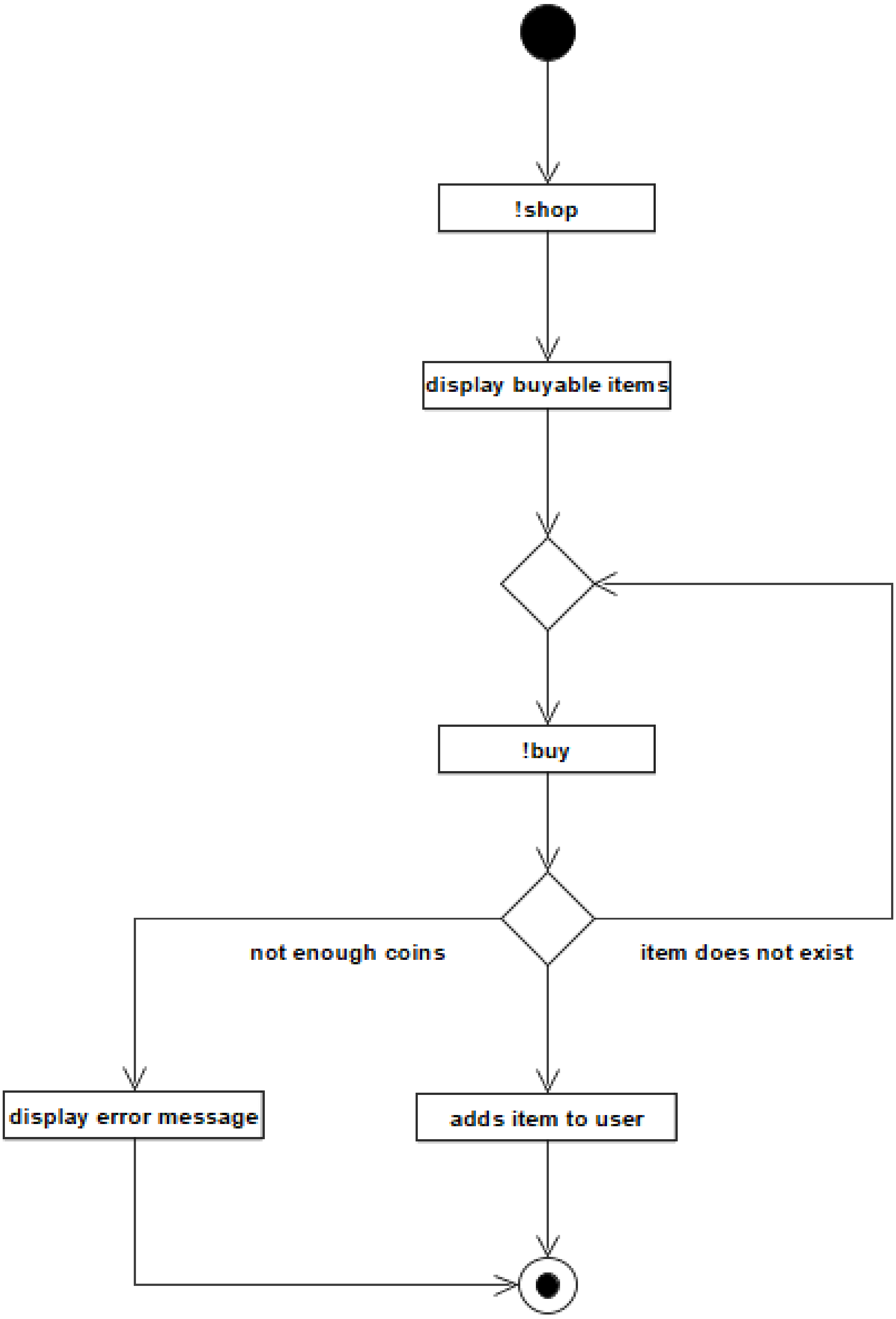
Input Field	Valid Inputs
Text bar	False Prefixes or wrong spelled Commands

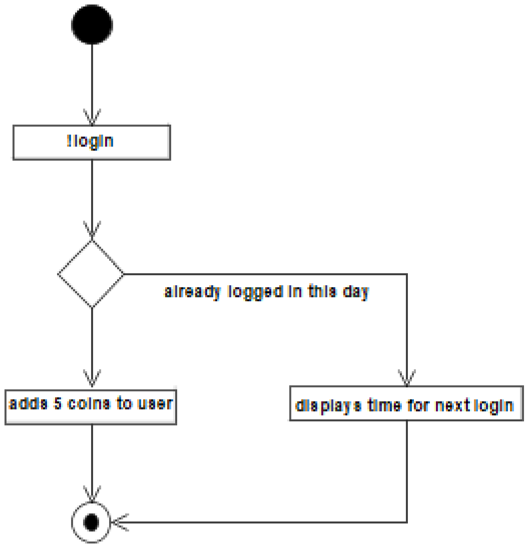
2.2.7 Workflow

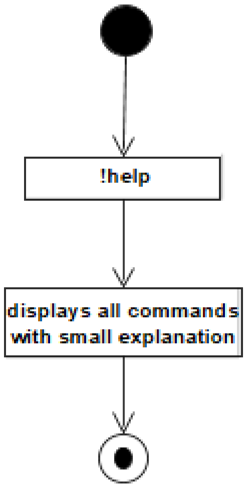


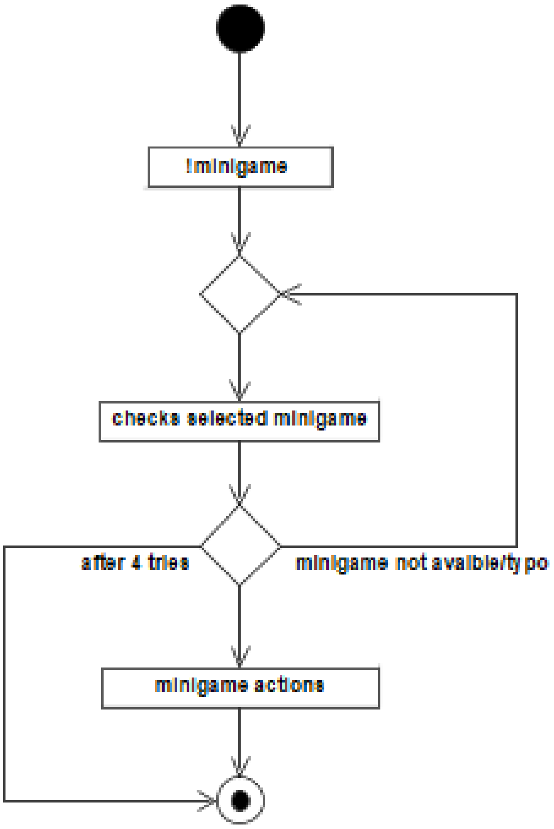

!play
joins the channel
checks link
if user is in no channel
if not valid
else
after 4 tries
process command
after 4 tries
else
else
if error occours
plays music
leaves channel
display output

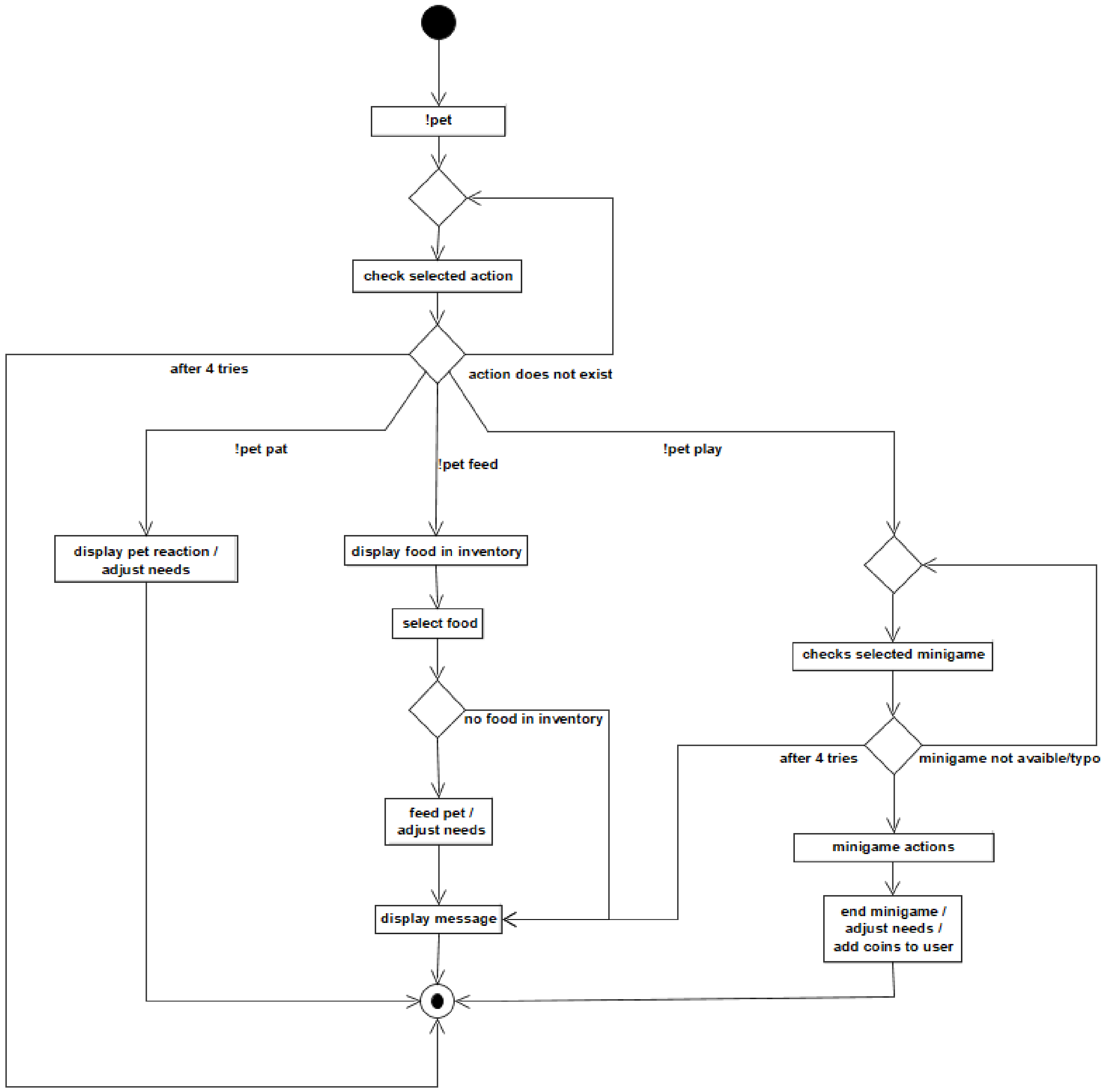

!userprofile
checks input
after 4 tries
if not valid
username avaible
else
user exists
display error message
display user profile
display executors profile


!shop
display buyable items
!buy
not enough coins
item does not exist
display error message
adds item to user


!login
already logged in this day
adds 5 coins to user
displays time for next login


!help
displays all commands with small explanation


!minigame
checks selected minigame
after 4 tries
minigame not avaible/typo
minigame actions



3. Non-functional Requirements

ID:	NFR_001
Name:	Use Command
Type:	EFFIC
Description:	The respond time of the commands should be under 2000ms in the best case, depends on the internet connection and the command which is used
Assigned use cases:	<002>

ID:	NFR_002
Name:	Simplify Commands
Type:	USE
Description:	The commands should be as simple as possible
Assigned use cases:	<002>

ID:	NFR_003
Name:	Language
Type:	MAINT
Description:	The bot and the commands should be in English
Assigned use cases:	<001, 002>

ID:	NFR_004
Name:	Legal Constraints
Type:	LEGAL
Description:	The links for the music function should be legally available for the country where the bot is running
Assigned use cases:	<002>

ID:	NFR_005
Name:	Add/Remove Commands
Type:	MAINT
Description:	If we have to add or remove some functions of the bot, we have to shut it down. While the bot is offline, other users can’t use it
Assigned use cases:	<002>

Types of non-functional requirements

Type	Name	Description
USE	Usability requirement	This requirement is to make the target group as described in section 1 is liking to work with that system.
EFFIC	Efficiency requirement	Run-time and memory efficiency. What are the constraints under which the system has to run
MAINT	Maintenance and portability requirement	Which maintenance or porting effort is expected in the future? Internationalization expected? Porting to different hardware platform? …
SEC	Security requirement	Security requirements comprise confidentiality, data integrity, and availability. How much do we have to consider that data is not accessible to unauthorized persons? Is the correctness and/or consistency of data to be guaranteed? How severe are total system faults?

Type	Name	Description
LEGAL	Legal requirement	Are there any standards or legal constraints to be considered?

4. Quantity Structure

We will use a Database because we have to store how many coins each user has, the different memes for the meme function, the items which one has bought in the item shop. We also have to store the current state of the pet one has, for example if the pet is hungry, tired or dirty.

5. System Architecture and Interfaces

The Bot has many functions such as “!play”, “!pet” or “!meme”. The respond time should be under 2000ms. The commands are in English and should be as simple as possible. For example, “!playmusic” is much more complicated than “!play”. The links for the Music function will be checked, in order to prevent legal complaints. We are able to shut the bot down, if we have to maintain it or add/remove functions.

6. Acceptance Criteria

<AC_001> - <The bot should recognize a command if one was typed in>

Test Step	Expected Behaviour	
A user types in !play	The message should be noticed as a command. If the command is valid then the bot should execute it. If the command is not valid the bot should display an error message.	

<AC_002> - <The most important functions should work (!help,!play)>

Test Step	Expected Behaviour	
A user types in !help/!play	If !play was typed in the bot should join the channel where the user is and play the requested music. If !help was typed in the bot should display a list where all function which are implemented are listed with a small description for each function.	

<AC_003> - <The bot should be at least 70% of a day online>

Test Step	Expected Behaviour	
If a user types in any command in order to use the bot	The bot should be online and execute the command	

7. References

https://discordbots.org/bot/memes
https://avairebot.com/
http://steambot.site/
https://mayawinterfox.com/
https://tatsumaki.xyz/