



College of Engineering,  
Forestry, and Natural Sciences

Tanner Massahos, Joseph Remy, Julian Bell, Tyler Boice, Chase Mosteller

Title of Project: roBOTically efficient

[GitHub](#)

D.2 Inception – Due: 2018-02-23

CS386 – Software Engineering – Spring 2018

Dr. Marco Gerosa

## 1. Positioning

### 1.1. Problem statement

The problem of task management when also wanting to play video games affects colleges students and those with little free time. This commonly leads to sacrificing one for the other, be it not completing tasks in order to play video games or vice versa.

### 1.2. Production Position Statement

College students and those with little free time may feel as if they are pulled away from what they enjoy by what they need to do. Our application, *roBOTically efficient* aims at assisting those individuals, specifically those who play 2007Scape, in having the best of both worlds by accomplishing both their IRL goals and in-game goals. *roBOTically efficient™* is an application which contains many bots that will do all the boring/hard work for these individuals who have a hard time balancing playing video games and accomplishing tasks. Our application differs from others on the market because we are attempting to make ours in such a way which they are undetectable by the creators of the game by replicating human movements perfectly.

### 1.3. Value proposition

*roBOTically efficient* is an application jam packed with highly sophisticated scripts that when activated will play 2007Scape for you. This will allow you to accomplish tasks that are also important to you in the outside world while making the gains you would like in-game. By utilizing *roBOTically efficient* you can never fall behind in any aspect of life; that's why our motto is "no XP waste in any aspects of life". *roBOTically efficient* is not like other 2007Scape bots; it is virtually undetectable, due to it replicating human movements with a small degree of error, rather than the classic direct line movement many other bots naively opt into.

## 2. Stakeholders

- Our **users** will be the ones utilizing this application, so they are of utmost importance:
  - Old School RuneScape players (primarily students, teens, younger adults)

- Busy individuals from the above groups who want to spend less time on the game overall
- Not only busy people, but also those who just want to get to the fun part of the game without doing whatever they deem to be 'boring' (like training undesired/tedious skills)
- **Our development team**, as we are the ones developing the software
- **Jagex Ltd.**, the creators of Runescape, as they aren't necessarily for the utilization of bots
- **PowerBot**, a competitor that also makes Old School RuneScape bots

### 3. Functional requirements

1. Create a GUI for the application - makes it easier for the user
2. Create a bot for fishing - most people dislike fishing so we will create a bot that does it for you
3. Create a bot for woodcutting - most people dislike woodcutting so we will create a bot that does it for you
4. Create a bot for mining - most people dislike mining so we will create a bot that does it for you
5. Create a bot for firemaking - most people dislike firemaking so we will create a bot that does it for you
6. Create bots that are human like - we don't want the account to get banned so we will make them as human like as possible
7. Create a bot that responds to chat nearby - gives it the more human feel

### 4. Non-functional requirements

1. Reliability - This bot should be trustworthy and work easily.
2. Undetectability - Jagex should not be able to determine that this bot is in use.
3. Security - The user should feel secure with their account while utilizing the bot.

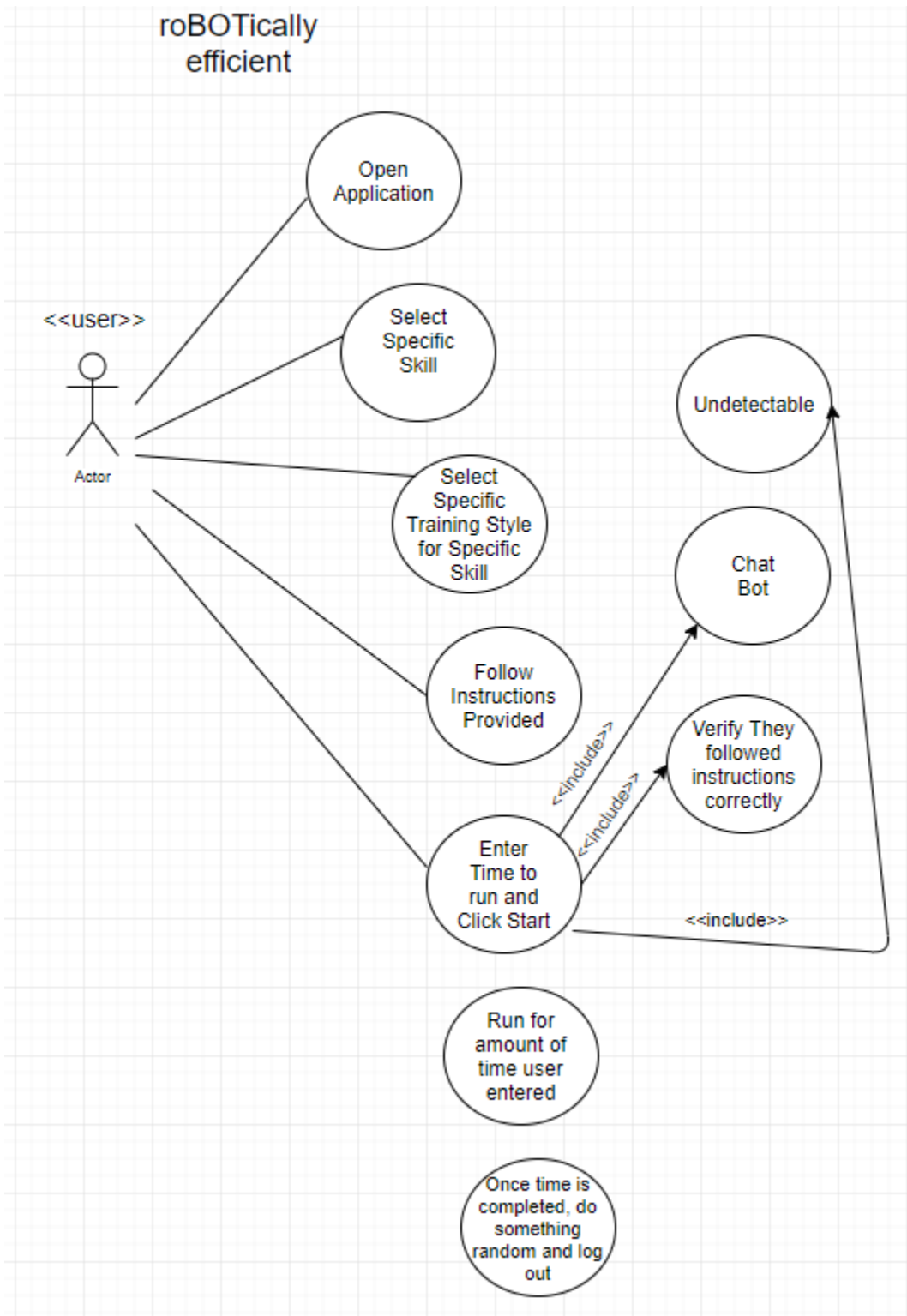
4. Availability - The bot should be easily accessible online for all users.
5. Cost - The bot should not cost anything.
6. Modifiability - The bot should be easily modifiable by other developers.
7. Understandability - The bot should be easy to learn and understand.
8. Minimal User Dependency - The bot should be instantly understandable and usable; that is, there should not be any high degree of complexity, especially at the start of the application.
9. Usability - The bot should be easy to use from start to finish.

## 5. MVP

For the minimal viable product, we are going to create the GUI with just one bot for one skill, and make sure that it satisfies both the functional and non-functional requirements. We will validate this by testing it ourselves a few times on a few different accounts. Once we have our first skill created, adding other skills later to provide more functionality should be fairly straightforward.

## 6. Use cases

### 6.1. Use case diagram



## 6.2. Use case descriptions

### Tanner Massahos

The user will open the application, upon opening the application the user will select from a screen of icons that align with what skill they would like to bot. They will click on the skill they would like to bot, from there they will receive a list of ways in which that skill can be bot, they will select their preferred method and it will display instructions on what they need to do for the bot to run perfectly, they can also enter in how long they would like the bot to run. Once they have set up their account in the required area, they will click go and the software will verify they are in the correct location, if they are not it will stop running and display an error to the user. If they were in the correct spot it will run based on however long they entered. Once completed the account will do something random and log out.

### Joseph Remy

First, the user will install the application and its requirements. Next, the user will open the application and be greeted by a window, which allows the user to do several things. Primarily, the window lists skills that the bot can accomplish. The user will select a skill and the window will transition to a new view. This shows the instructions for how to set up their account. Once they are in the correct spot, they will go back to *roBOTically efficient* and say yes to a confirmation box. The application will confirm then run until it is interrupted by an action (shutting off the computer, loss of Internet, keyboard interrupt, etc.). If the user account is NOT set up properly, the application will give an error and wait for the user to try again or cancel.

### Tyler Boice

The user opens the application and will see a window pop up. This window acts as the client for *roBOTically efficient* and will then load into the game of Runescape. From there the user will see the user login window and they will have a host of options at the top for various skills and money making tasks that the bot can perform. By logging into the game and then clicking one of the options the user will be greeted with an options screen for the particular bot which lets them further customize the certain actions the

bot takes while training skills or making money. The bot will perform the specified actions until the user tells it to stop or the program is interrupted. If the bot incurs any errors while running or cannot perform the tasks that it was given it will alert the user and ask them to reconfigure the tasks they wish to perform.

### **Chase Mosteller**

First the user will run the application. They will then enter in options in the user interface including the type of training they would like to do, their username/password for logging in, the options for that training and an optional goal time for the application to run. If the account is in a valid area to train, the bot will run. When the conditions aren't met to continue, such as running the application in an invalid area, the bot will attempt to log the user out and crash.

### **Julian Bell**

The user opens *roBOTically efficient*, and selects the icon detailing which skill they want to train. Upon selecting the skill and training style preferred for that skill, the bot displays a prompt on how long it should run for and where the user should go to set up the bot to run properly (if this is not satisfied, the bot throws an error message letting them know they're not in the right place). After the user is in the right location and hits the "go" button, the bot process begins and continues through the specified time amount. The account performs a random, 'human-esque' action when time is up and logs off of the game.

## **7. User stories**

### **Tanner's User Stories**

- 1) As a user, I want to get my skills leveled up in 2007Scape so I can play the fun parts of RuneScape.
- 2) As a user, I want to get gold from botting, so that I can buy the items I want.

### **Joseph's User Stories**

- 1) As a college student, I want botting software so that I can progress through the dull and boring parts for the more interesting locations and quests.
- 2) As a RuneScape player that wants to make gold during events like Double XP Weekend, I need a bot to help me increase my skills and farm for materials.

#### **Tyler's User Stories**

- 1) As a person with a busy job, I want a bot that will allow me to maximize my time away from the computer by levelling through the more boring aspects of 2007Scape.
- 2) As a 2007Scape player with a max level account, I want a bot that will make money for me while I play other games, since I do not need to skill, but money is always useful to me.

#### **Chase's User Stories**

- 1) As a college student, I want to be able to make progress in some of my mmorpg games, but also be able to spend enough time studying or completing homework.
- 2) As a Venezuelan citizen, I want to be able to make a profit off this game to support my family.

#### **Julian's User Stories**

- 1) As a college student, I want undetectable bot software that lets me get to the non-tedious parts of the game I really want to play without using up a large chunk of my free time.
- 2) As a RuneScape user, I want to get ahead in valuable game skills to make in-game money, so I can buy more items and accessories for my character.

## **8. Trello**

Chase set-up the Trello board and team.

Public Link to Board:

<https://trello.com/b/4hvAsbj3/naucs386opdingo>



Team Page:

<https://trello.com/operationdingo1/>

## **9. Group participation**

Tanner Massahos – Did his user stories, user description, created user case diagram, also typed up functional and non-functional reqs (25%)

Joseph Remy – Typed up 6.2 paragraph, 2 user stories in section 7 (18.77%)

Julian Bell – Typed up 6.2 paragraph, 2 user stories in section 7 (18.76%)

Tyler Boice – Typed up 6.2 paragraph, 2 user stories in section 7 (18.75%)

Chase Mosteller – Typed up 6.2 paragraph, 2 user stories in section 7 (18.72%)