CEN 4010 Principles of Software Engineering, Fall 2021

Milestone 1 Project Proposal and High-level description

Group Number 5

Team Lead:

 Chelsea Morgan ([cmorgan2016@fau.edu](mailto:cmorgan2016@fau.edu))

Team Members:

Saad Bhullar ([sbhullar2018@fau.edu](mailto:sbhullar2018@fau.edu))

Sesel Brown ([seselbrown2017@fau.edu](mailto:seselbrown2017@fau.edu))

Omar Azad ([oazad2016@fau.edu](mailto:oazad2016@fau.edu))

Evgeniia Maksimova ([emaksimova2020@fau.edu](mailto:emaksimova2020@fau.edu))

# Executive Summary

Habitmon is a tool to let a user gamify their self-improvement journey while meeting others and becoming worldly. The user upon signing up can pick preset goals, tasks and habits or make their own and choose a few of the dev-team run groups where they can share or view other’s progress. Once they complete a certain number of goals, tasks, and habits they will earn a spin which will give the user a random critter. This critter can grow, and fight based on the user completing their goals, tasks, and habits.

Users can compete against each other via Group made challenges or by challenging each other. This is done by seeing who makes the most progress, improved the most and seeing who can meet goal posts first.

# Competitive Analysis

## Competitor 1: Habitica

|  |  |
| --- | --- |
| **Habitica** | **Habitmon** |
| Chat/Guild/Party implemented |  |
| Add, edit and complete tasks |  |
| Character Customization |  |

## Competitor 2: Habit Hunter

|  |  |
| --- | --- |
| **Habit Hunter** | **Habitmon** |
| High game content and low productivity content | An even mix of both game and productivity content |
| Have to design own tasks, habits and goals | Can use prebuilt tasks, habits and goals |
| Completing tasks, habits and goals gives in-game currency | Completing tasks, habits and goals gives xp and in game items |

# Data Definition

**Game Title:**

* Betterment Beasties
* Motivational Monsters
* Task Monsters
* Taskmasters
* Habits and Habitats

**Creature Names:**

* Beasties
* Critters
* Monsters
* Deviants

**User:**

* Does not have a character, can pick image avatar
* Can pick from custom and premade tasks, habits and goals
* Is given 1 creature after do so
* Can choose which dev run groups to join which have several weekly challenges
* Can assign tasks and habits to a certain creature, if not assigned benefit is assigned among all who can use it
* Can use creatures in battles

**Goals:**

* Long term projects broken down into milestones
* Completion gifts a random critter
* Can be used as a challenge to other users, where you can complete and compare progress together

**Habits:**

* Non daily, recurring events users wish to incorporate into their life more
* Single entity, no milestones
* Gifts…….benefits to monsters? In game currency?

**Tasks**

* Daily events users need to do each day
* Gifts xp to one or all monsters
* Not completing can cause health use
* Can be put into categories to increase certain creature traits

**Challenges**

* Need to enter 1 creature to join
* User or group run
* Users compare tasks, habits and milestones completed within a week
* Will gift prizes based on progress

**Creatures**

* Have different rarities
* Can only level up and strengthen by completing benefits(?) and tasks
* Can be used to join challenges or battles
* Have habitats
* Have several traits that can be improved with xp

**Habitats**

* Homes of creatures
* Can be decorated and upgraded by user
* Upgrades and decorations come from battles and crafting

**Battles**

* General turn based battles
* Limited times to battle per day
* Gifts upgrades and decorations

**Crafting**

* Habitat decorations and upgrades can be made here
* Certain number of materials are needed for crafting
* Materials earned by forging

**Forging**

* Creatures can be sent out to gather wood, ores and other materials
* This uses idle game mechanics
* Various traits creatures have and can increase with increase the amount found

# Overview, Scenarios and Use Cases

# Initial list of high-level functional requirements

The following is a list of functional requirements that we plan the implement for our project. This is the initial list and additional functional requirements will be provided later.

**User Story:** As a new user would like to create an account

**Functional requirements:**

1. The system/application must allow the user to create an account with a username and password
2. The system/application must allow the user to create an account with Google or Facebook accounts
3. The system/application must display message if username is already in use

**User Story:** As an existing user, I would like to be able to log into my account

**Functional requirements:**

1. The system/application must allow user to log into account by entering their email and password
2. The system/application must allow user to log into account with Google or Facebook accounts
3. The system/application must allow user to change/ reset password

**User Story:** As a user, I would like to create an avatar

**Functional Requirement:**

1. The system/application must allow user to select a gender
2. The system/application must allow user to select a body
3. The system/application must allow user to select hair
4. The system/application must allow user to select clothing

**User Story:** As a user, I would like to read messages for general updates, world events, game events and donations

**Function Requirement:**

1. The system/application must allow the user to read messages (updates, world events, game events and donation)

**User Story:** As a user, I would like to receive a reward for completing goal(s)

**Function Requirement:**

1. The system/application must provide the user with a reward

**User Story:** As a user, I would like to choose my main group

**Function Requirement:**

1. The system/application must provide a mechanism to choose the main group

**User Story:** As a user I would like to make purchases, payments and/or donations

**Function Requirement:**

1. The system/application must provide a mechanism to make purchases, payments or donations
2. The system/application must provide secure means to make purchase

**User Story:**

**Function Requirement:**

# List of non-function requirements

The following is a list of non-functional requirements that we plan the implement for our project. This is the initial list and additional non-functional requirements will be provided later.

**Non-function requirement:**

The system/application shall provide a user login page

**Non-function requirement:**

The system/application shall provide the user the ability to enter a username and password

**Non-function requirement:**

The system/application shall provide the user the ability to login with a Google or Facebook account

**Non-function requirement:**

The system/application shall provide the user the ability to create a new account with a username and password

**Non-function requirement:**

The system/application shall provide the user the ability to create a new account with an existing Google or Facebook account

**Non-function requirement:**

The system/application shall identify user at login

**Non-function requirement:**

The system/application shall create a unique ID for new users

**Non-function requirement:**

The system/application shall user the ability to create an avatar

**Non-function requirement:**

The system/application shall allow the user to pick gender, body, hair, clothing for avatar

**Non-function requirement:**

The system/application shall provide the user a reward method

**Non-function requirement:**

The system/application shall provide the user a secure method of making payments

**Non-function requirement:**

The system/application shall be responsive

# High-level system architecture

The following is the list of the high-level system architecture

Programming Language:

* C++
* Java Script
* Java

Database:

* MySQL

APIs:

* **Vulkan** – which is a graphics and compute API the provides high-efficiency cross-platform access to modern GPUs used in variety of devices
* **OpenGL|ES** – royalty-free cross-platform API for rendering 2D and 3D graphics which include consoles and phones
* OpenCL – royalty-free standard for cross-platform, parallel programming

# Team

## Initial roles:

**Product owner** **& Team lead** - Chelsea Morgan

**Scrum master** **& Back End Developer** - Evgeniia Maksimova

**Back End Leader & GitHub master -** Sesel Brown

**Front End Leader** - Omar Azad

**Front & Back End developer** - Saad Bhullar

# Checklist

a)   Team decided on basic means of communications - DONE

b)   Team found a time slot to meet outside of the class - ON TRACK

c)   Front and back end team leads chosen - DONE

d)   GitHub master chosen - DONE

e)   Team ready and able to use the chosen back and front-end frameworks - ON TRACK

f)    Skills of each team member defined and known to all - DONE

g)   Team   lead   ensured   that   all   team   members   read   the   final   M1   and

agree/understand it before submission - ON TRACK