


1. Process Deliverable: RewardBot Prototype

Link: <https://miro.com/app/board/uXjVLM1xxy4=/>

a. RewardBot Login Screen

RewardBot Login

RewardBot

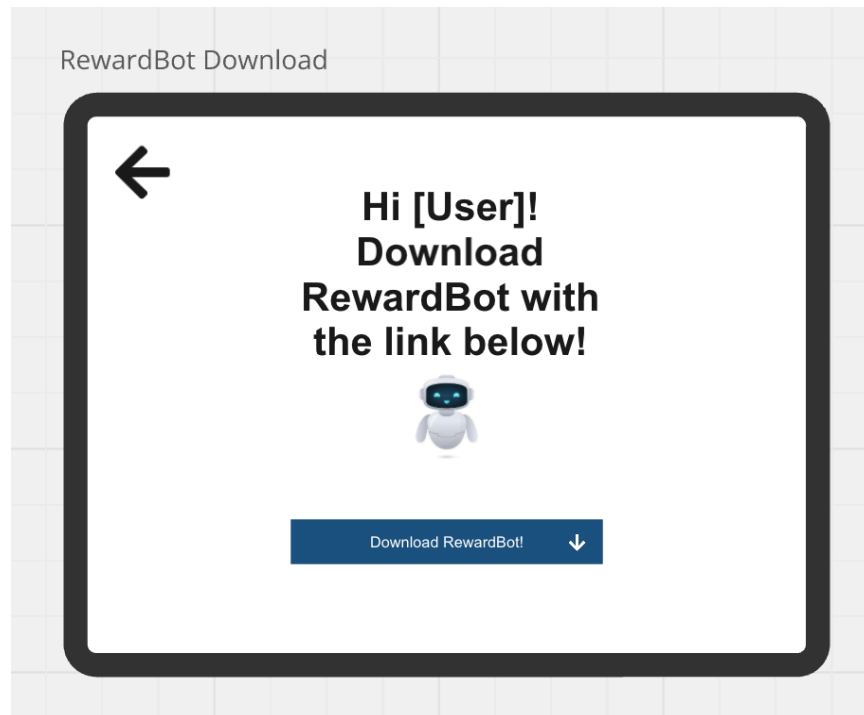


email

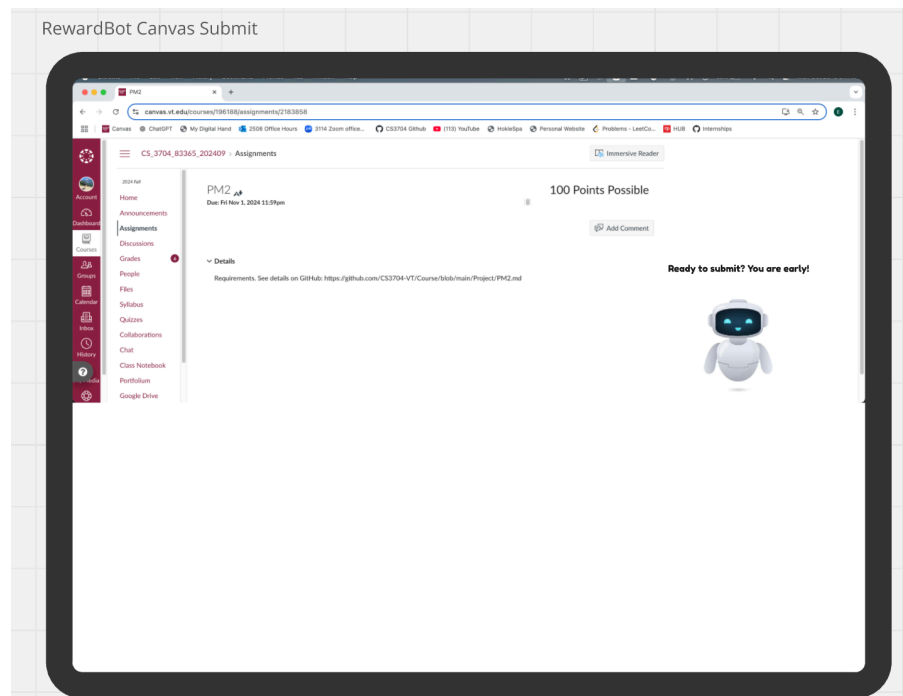
Login

Create Account!

b. RewardBot System Download



c. RewardBot integration with Canvas Learning Management System



2. Requirements Analysis

Based on the results of our groups' Requirements Elicitation in PM1, we were able to come up with five hypothetical functional and non-functional requirements for the RewardBot system:

a. Functional Requirements:

1. User Authentication should keep users secure

The RewardBot System should allow its users to login to their RewardBot account through their Learning Management System, like Canvas LMS, so that the system can be integrated into the students' respective LMS.

2. System should offer a variety of rewards

From our requirements elicitation, our survey respondents noted a variety of different desired prizes, which led us to pick out some rewards from the data, as well as some new ideas we came up with. The list of rewards we chose is: bonus points, food voucher coupons, digital badges, and extra "bank-days," which allow the student to submit an assignment a number of days late, withdrawn from their bank-day total they have. Some rewards would be rarer than others.

3. System should provide an Institutional Data Report

The system should report data to their instructors, so that they can see which students are succeeding, what the students are struggling with, and overall trends. Having this data provided would be critical in improving student comprehension, as instructors can then tailor their lesson plans to focus more on the areas that the students are consistently struggling with.

4. The system should keep user data secure

The system should implement measures to protect the users data from being exposed. Information like names, ages, emails, etc. must be kept private so there is no malicious activity.

5. The system should provide useful documentation

The downfall of many systems is due to poor UI. While certain designs can be implemented to reduce confusion, there will almost always be a user who does not know how to navigate the system. Thus, there must be proper documentation that is easily accessible for the users.

b. Non-Functional Requirements

1. Usability: The interface of the system must be easy to navigate

The interface for both the download and the integrated system must have a simple and intuitive interface. All of the possible actions should be clearly labeled, with minimal instructions to complete the activity. The tool should also be compatible with both mobile and desktop versions, as students often use both of these versions of their LMS.

2. Readability: The system must use consistent terms across all platforms, reducing user error as much as possible

The system should use the same icons for rewards, eg. the icon should be the same for all bonus point awards, icon awards, etc. That way, there is no confusion on what type of award was just given. Additionally, the system should provide documentation/instructions on how to use the system, as well as have a legend of the icons displayed when a reward is presented.

3. Performance: The system must be able to handle millions of requests per second for every student that uses the system

The system should be able to work across all of its users' devices at once. To accomplish this, the first step is to be able to handle all of the requests of its users (be able to reward all necessary users at the right time). Additionally, these requests should have a fast server response time, ideally less than one second per request.

4. Supportability: The system should work across all web browsers and device types

The system must be able to provide the functionality of rewarding its users on all versions that it supports eg. work on web and mobile versions. Students oftentimes utilize multiple devices to access their LMS, including their mobile phones and their laptops. Because of this, the RewardBot system must be able to work seamlessly between all platforms.

5. Implementation/Constraints: System must prioritize the privacy of the users' data

The last thing we want our system to do is cause an inconvenience for our users. If somehow, we were responsible for a data leak, it could lead to catastrophic issues, as people often reuse passwords on many different accounts on different services. This breach could cost the company lots of money, so it is crucial that all of this data is properly encrypted and safe.

3. Requirements Specification

User Case 1:

As a *Procrastinating Student*, I want to *use the reward that I have collected* to get the extra bank days to *submit the project late with no penalty*.

- **Given** Procrastinating Student has student permissions and has done a fair amount of reward from past projects
- **When** Procrastinating Student clicks on the “Redeem Reward” button
 - **And** clicks “Extra Bank Days”
 - **And** chooses the specific assignment to extend the deadline
- **Then** the system sends the request email to course instructor and extends the deadline by approval

Subtasks:

- Implementing the redeem function / UI for the award (4 points)
- Implement the request email function (2 points)
- FUNCTION POINTS: 6 points

User Case 2:

As a Course Instructor, I want to find a list of students that are failing my class so that I can have some insight to improve my course structure.

- **Given** Course Instructor has Instructor permission
- **When** Course Instructor clicks student status report
 - **And** filters students by course failing grade
- **Then** the system finds a list of students with their grade status

Subtasks:

- Generate the grade report (1 point)
- Add the function to filter the grades (1 point)
- FUNCTION POINTS: 2 points

User Case 3:

As a New Student User, I want to see the instructions on how to use the system to *spend a motivated and effective semester*

- **Given** New Student User has a student permission and a new user permission
- **When** the New Student User logs in
 - **And** clicks “Tutorial” in the popup only shown for new users
- **Then** the system marks the student has a non-new user, and leads the user to the tutorial documentation.

Subtasks:

- Implement the popup for the tutorial when there is a new user (2 points)
- Make tutorial content (1 point)
- Track if the user is new or returning (1 point)
- FUNCTION POINTS: 4

User Case 4:

As a Teachers Assistant, I want to *be able to access students assignments to reward eligibility and assignment progress to assist with tracking performance.*

- **Given** the Teacher's Assistant has TA permissions
- **When** the TA clicks on "Student Assignment Access"
 - **And** selects a specific student and /or assignment
- **Then** the system displays the student's assignment progress and rewards earned, allowing the TA to provide assistance based on their progress

Subtasks:

- Implement reward eligibility for students (3 points)
- Display the students progress (1 point)
- Make it so that the TA have their own section (above student but less than professor) (2 points)
- FUNCTION POINTS: 6