**Raspberry PI Check-in & Checkout System**

**Spring-2020**

**Group:** NIMPS Engineering

**Team Members:**

**N**icholas Economou, **I**saiah Nwukor, **M**arco Martinez, **P**iyush Dahiya, **S**tuti Rana

**Team Capabilities Description**

|  |  |  |  |
| --- | --- | --- | --- |
| **Team Member Name** | **Computer Science Education** | **Computing Skills Inventory** | **Work Experience** |
| **Stuti Rana** | Bachelor’s of Computer Science Graduating December 2020 | Java, Python, JavaScript, Focus on mainly Data Science, MySQL, Tableau, TensorFlow | Using Big Data Analytics as a junior consultant at Mather Economics  Volunteered as a Mentor for Girls Who Code teaching middle school girls basic coding skills |
| **Isaiah Nwukor** | Bachelor’s of Computer Science Graduating May 2020 | Full Stack Development, Databases  Liquid, CSS, Javascript, PHP, MySQL | StoreMods: Shopify Developer  Sideways 8: Frontend Web Developer  Applied Imagination LLC: Frontend Web Developer |
| **Nicholas Economou** | Bachelor’s of Computer Science Graduating May 2020 | Python, Java, C, UNIX Shell Programming, MySQL, CSS Grid, Databases | Language Research Center: Implementing Raspberry Pis  LogicBox: Intern programming fundamentals with Python |
| **Marco Martinez** | Bachelor’s Of Computer Science Graduating August 2020 | Java, Python, JavaScript, Typescript, C#, HTML, CSS, MySQL, C, Angular, Django | FoundationsIT: Software Developer  Projects: Fixit Website(Angular)  Web Scraping Application (Python) |
| **Piyush Dahiya** | Bachelor’s of Computer Science Graduating May 2020 | Python, Java, UNIX, HTML, CSS | Working with GSU Network system Admin: Aug 2016 to present.  Projects: library management system in Java and c, machine learning implementations with python across a multitude of data sets |

**Planning & Scheduling: Work Breakdown Structure**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Assignee**  **Name** | **Email** | **Duration** | **Dependency** | **Task** | **Due Date** | **Note** |
| Stuti Rana | srana1@student.gsu.edu | 3.5 hours | Collaborating With Piyush On Refining System (Task 3) & with Marco for (Task 5) | Work On Revising & Refining System (Task 3) System Modeling Analysis (Task 5) | 02/20/20 | **100%** |
| Isaiah Nwukor | Inwukor1@student.gsu.edu | 3 hours | Depending on team members to complete their tasks, and organizing them in the shared google drive to create the report. | Scheduling meeting times, assembling the final report, creating work distribution task list on Github (Tasks 1,2,6) | 02/20/20 | **100%** |
| Nicholas Economou | Neconomou1@student.gsu.edu | 3 hours | Collaborating with Marco on System Requirements (Task 4) | Work On System Requirements (Task 4)  Record Group Video/Upload to Youtube (Task 7) | 02/20/20 | **100%** |
| Marco Martinez | Mmartinez26@student.gsu.edu | 3.5 hours | Collaborating with Stuti (Task 5) and with Nicholas (Task 4) | Work on System Requirements (Task 4)  Work on System Modeling (Task 5) | 02/20/20 | **100%** |
| Piyush Dahiya | Pdahiya1@student.gsu.edu | 3 hours | Collaborate with Stuti On Refining System (Task 3) and proofread/ Quality Check the Report (Task 6) | Work On Revising & Refining System (Task 3)  Proofread/ Quality Check the written report (Task 6) | 02/20/20 | **100%** |

**Teamwork Basics**

**Work Norms:**

Work will be distributed evenly based on the group members’ strengths. Where the group members have weaknesses, they can participate in observing. The team leader will set deadlines at the beginning of the project according to everyone’s schedule and when the project is due. If a group member slacks off the group will talk to the person directly to make them understand that we are a team and if they slack off the team cannot function to the best of its ability. Hopefully, after this talk, they understand and begin to pull their weight, but if not, then this will be reflected in their final grade and how their work is reviewed at the end of the assignment. We have to understand that in this randomly chosen group, everyone will have different personalities and thus differing opinions and work habits are to be expected. However, we must work around these difficulties and communicate openly as a group to reach a general consensus.

**Facilitator Norms:**

Whoever is chosen as a team coordinator will also act as the facilitator. We will not use a separate person to be the facilitator. If the team gets off track anytime during the project the facilitator will refocus the group and make sure that everyone in the group is participating and stay on top of their role by giving gentle reminders. Make sure that everyone in the group is sticking to the time frame that was agreed upon at the beginning of the project.

If the team runs into a roadblock the facilitator should give different suggestions as to how to solve the problem. If team members are having problems the facilitator should act as a mediator/neutral ground to help walk through these problems and encourage communication. Lastly, the facilitator should give a quick summary at the end of the meeting to ensure that everyone is on the same page.

**Communication Norms:**

The majority of the communication will take place on Slack. Everyone in the group has downloaded the Slack app so that we can easily receive notifications. We also have traded individual numbers since texting is definitely a popular/convenient form of communication. So far, it seems that everyone likes to communicate through Slack and no one has suggested a different preference. We have also made a Google doc so that everyone can have access to the written parts of the assignment and add feedback/make changes. For future group assignments, we will likely stick to this form of communication.

**Meeting Norms:**

The team coordinator is responsible for coordinating the meeting times. Although some times are conflicting the group members know their responsibilities and need to dedicate time to this project. A good place to hold meetings is the study rooms in the library. We just have to make sure to reserve the space. Most members prefer to meet after class. If someone is late to a meeting or misses a meeting it could hold up the whole group. The other members of the group must make sure to communicate with this person to understand what the reason is that they are missing meetings and see if there is a better time for them to meet.

**Consideration Norms:**

People can eat in meetings but not smoke, Their behavior should not be distracting to others and if it is they need to stop and change their behavior. Norms can be changed with communication and openly expressing problems and solutions with the rest of the team.

**Hints for handling difficult behavior:**

**Overly talkative-** This person is usually someone who wants to let everyone know how much they know or someone who just loves to be at the center of attention and doesn’t know how to tone it down to let others have a chance to speak. This person could also be the “class clown” type who doesn’t know when to stop joking and talking. This type of behavior can be distracting and cause the group to veer off-topic. This person needs a gentle reminder from others in the group that while it’s great that they are so eager and excited, everyone deserves a chance to speak and be heard.

**Too quiet-** Maybe there is a person in the group that is the opposite of the overly talkative type and is shy/ introverted and needs to be drawn out of their shell. It’s important to the group’s performance that everyone’s opinion/ideas are heard since being in a team everyone needs to work together. This person could have a really good idea, but no one would know because they are too timid to voice their opinion. The group needs to make a special effort to let this person know that they are valued and that their ideas matter. The group could directly ask this person- “Hey, what is your opinion/idea?” to get them to share his or her thoughts. From past experience, there was this girl that I worked with on a team before who was extremely shy and would not talk during meetings. Everyone thought that she would hurt the group but in the second meeting we had she showed up incredible ideas and models that she had been working on. She was just quiet but she had a lot of ideas. We fixed this by talking to her one on one. We found out that she had anxiety in large group settings.

**Argues-** This person is critical of everything in the group and likes to argue and create negative tension in the group. The group should again try to open friendly lines of communication with this person to understand what their issue is and if it can be handled in a positive way that is not argumentative. Arguments create unnecessary stress within groups and sway the group from its main focus.

**Complains-** This person could have some sort of pet peeve or just like to complain in general. The group has to listen to the member’s complaint and set aside time to discuss it. Problem-solving is part of the project this semester and a part of learning how to work in a group setting.

**Hints for Handling Group Problems:**

In addition to problems with individual group members, there could also be problems that the team has to face altogether. Some examples of these problems include floundering, going off on tangents, making a decision too quickly, not making a decision, feuding between group members, ignoring or ridiculing others, and group members not pulling their own weight and doing the work. We believe that within all of these problems the overall solution remains the same. Everyone can solve these problems through open and active communication. If a member continues to be difficult it will negatively impact their personal grade when it is documented in the planning and scheduling table.

There was once a project I worked on last semester in my Web Programming class and it just seemed impossible to meet up with one member. He got his work done but his commentary and code were sometimes difficult to understand. When we were implementing the project we finally solved the problem by face timing him and having him walk us through. So even though he wasn’t physically present at the meeting he still was able to contribute and make sense of his work.

**Problem Statement**

Our project aims at optimizing GSU’s check-in and check-out system, to automate functionalities for ease of use, and to ensure a better system for keeping track of items that are checked out and checked back in. Requirements were elicited by setting up a meeting with our instructor coordinator and the network admin. On our team we have some people familiar with raspberry pi’s and also have some understanding of the current check-in/checkout system that has given us more details about what may be needed for the app. Our project is focused on Teacher assistants at the GSU library where they offer technical equipment rental for students to check out.

Our project solves the inefficient and insecure paper system that is currently being used to keep track of inventory and student rentals. It will also be able to tell if a student has not returned an item back in a timely manner and will alert attendants instead of having to manually search the excel spreadsheet.

There are few direct alternatives to solve the problem we are aiming to solve, there are some third-party applications that perform similar tasks as we aim to do but none are quite as specific as we are aiming to be. This project can help to streamline GSU’s check-in and check-out system for Raspberry Pi’s and keep track of equipment inventory.

The main objectives of the project are to provide TA’s, specified instructors and the network admin with a system that tracks the status of the raspberry pi inventory, and allows TA’s to seamlessly check-in and out equipment to lab students. With our system, we would digitize the initial checkout forms, the warning/disclaimer, and the TA check-in process. The system needs to be able to account for all loaned out equipment as well as their check-in condition, and date returned.

The system needs to report any damaged or overdue equipment so that GSU staff can replace and take action on a case by case basis. The target customers are GSU lab students, who need to rent the equipment the entire semester for their classes. What differentiates us from the standard check-in/checkout system is our approach nudges students to return items early. As far as the competitors, for the google play store, there were no direct competitors, but we did find some apps that are doing similar things for different industries:

**Check-in Scan**

<https://play.google.com/store/apps/details?id=com.checkinscansl.checkinscan>

Check-in scan is an app that helps landlords/Airbnb hosts simplify the check-in process and legal documents signing for tenants and guest renting their properties. They use digital signatures and have an admin backend for the property owners to manage their rentals.

**Check In Check Out Mobile**

<https://play.google.com/store/apps/details?id=com.ss.checkinout>

This app was created to help businesses keep track of who is currently in the office, and who is out for various reasons (sick, business meetings, etc). It gives companies the opportunity to take note of trends and possible patterns, and schedule meetings at times that work best for the majority of the parties involved.

**Handy Library**

<https://play.google.com/store/apps/details?id=com.handylibrary.main&hl=en_US>

Handy Library is an app that allows users to search for books at their own library and scan book barcodes to get more details. The app sends out reminders to return books that have been checked out to friends and exports a backup file of the books on record to excel.

**RasPi Check**

<https://play.google.com/store/apps/details?id=de.eidottermihi.raspicheck>

This app provides an overview of your raspberry pi’s status. It provides you with system information, runtime, disk usage, processes and allows you to send the raspberry pi custom commands and shutdown or reboot the system.

While it is not a requirement, a system like this could help give the network admin and instructor some more tangible feedback on why devices are failing over time. When dealing with non-functioning equipment TA’s can sending system diagnostics to the staff, so they can troubleshoot problems without meeting in person.

This app can be built using spring framework, and with the information that we have from our instructor coordinator, and some meetings with the School’s Network administrator, we should have enough resources to create a useful app.

**User Requirements:**

## **User Functional Requirements:**

1. The system is a check-in / check-out application for the GSU CS Dept.
2. Students should be able to check out Raspberry Pi devices at the beginning of the semester. (Form 1)
3. The TA must be able to acknowledge and ensure the correct check-out of the devices. (Form 2)
4. The students must be able to return the devices at the end of the semester. (Form 2)
5. The TA must be able to effectively check and complete the check-in process with the students (Form 2).
6. Potential Edge Cases: Student enrolls late or drops out early.
7. The Network System Admin (Power User) & Instructor (Dr. Mussa) must be able to view at any time (through generated reports or otherwise):
   1. The status of all the devices (in stock / checked out etc.)
   2. The status of conditions of all devices (damaged etc.)
8. The devices need to have a unique identifier which will act as a unique id for the app.
9. The students will require TA approval through Form 2 to complete the check-in and check-out process.
10. Students must present ID verification upon check-in and check-out which will be stored in the app.
11. Students must confirm the disclaimer that they are liable for materials returned in damaged condition, or materials only partially returned

***From The Instructor Coordinator/Product Owner:***

Before checking out the student needs to confirm the following disclaimer/warning:

The student listed is responsible for the safekeeping and prompt return of all items listed on this form as checked out. If one or more components, becomes damaged or lost, the student is responsible for providing a replacement. Please speak with your instructor on what items are suitable. Failure to return all required material by the return date listed on this form may result in but not limited to a failing grade of zero for this course, a hold on your student account, and any additional legal action Georgia State University may choose to pursue in accordance to the missing or damaged equipment.

Date: Signature:

Checkout: Your Raspberry Pi kit should contain:

Write the Raspberry Pi Serial#

Items: (Checklist record during checkout & during checkin)

● Raspberry PI 3 B+ with touchscreen

● Keyboard

● Power Cable

● HDM I Cable

● SD Reader

● Plastic Container

(Optional Notes)

Student Info Needed:

Name

Panther ID:

Lab Session: Lab Day, Lab Time

Email Address:

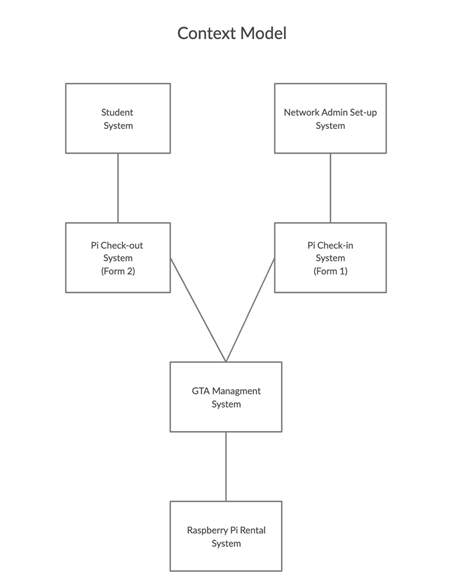
Checkout Date: ex: 01/31/2020

CheckIn/Return Deadline (During Last Lab Time - Lab 14 )

--------------------------------------------------------------

**System Requirements**

This context model is centered around a raspberry pi checkout system with the rental database system being the center of interactions. The student would need to register before interacting so that they are accountable and allowed to participate with checking out Raspberry Pi’s if they want to. The network admin will handle the devices and check them out to the graduate teaching assistant, (GTA), using a separate form for checking in. The mediator between checking in and checking out the devices will be the GTA. Only the GTA will be responsible for checking in and out raspberry pies. It will also be their duty to manage the Rental system and to check for items being returned are in a similar condition to when they were checked out.

****

**System Requirements:**

***Functional Requirements:***

1.) Download the app (or TA provides Tablet) and register a new account with my GSU email.

2.) The registration form asks for students’ name, panther ID, and lab session (day/time).TA goes through the process of listing checked out items, takes a photo of the kit, registers the device id under the students name and has the student digitally sign the warning/disclaimer.

3.) Once checked out the student receives the equipment, is told the return date, and is also emailed a link to the app listing its features, a reminder of what they checked out, and when they need to return their equipment.

4.) The app if downloaded will provide them with access to Raspberry Tutorials for setting up their equipment and provide technical information about their raspberry pi. If they run into issues, there is a button that can send the diagnostic details to the GSU staff along with a description of the issue to cut down on in person troubleshooting.

5.) A week before the final lab, the app sends a notification to the users email and through the app itself to show them a photo of their checked out equipment, where to return the device and all the things that need to be included.

6.) The student returns the device kit to the TA, and the TA takes account of all the equipment and notes the condition upon return to the app.

Function requirements, upon identification, directly impact the software to be produced not only in terms of specifications but also in terms of higher-level architecture. Since we are in the beginning stages of the development and based on what the client (Dr. Mussa) has advised yet, the use-case diagram is presented below. Our team has yet to have a chance to speak with Mr. Bryan, Network Security Admin, upon which the functional requirements and use-case diagram will be updated accordingly.

1. ***Non-Functional Requirements:***

These requirements specify the constraints on the system or the development process and are related to how the functionality is performed.

**-Product Requirements:**

* The database and all student info must be stored on GSU servers for security and dependability.
* The system must use encryption and appropriate encryption and keys (e.g. HTTP Secure).
* The system must be intuitive and user-friendly.
* The system must have a tiered level of access, i.e.:
* i) Students can only fill out Form 1.
* ii)TA verifies students Form 1 and then fills out Form 2 to confirm check-out.
* iii) Network Admin and Instructors can manage inventory, and generate reports

**-Organizational Requirements:**

* The product must be developed by May 2020 (end of class).
* The product must be available to use and view at all times by the users who have access accordingly.

**-External Requirements:**

* It must be ensured that student data (including ID pictures) is kept securely and in accordance with all relevant privacy and regulatory metrics.

***B) Functional Requirements:***

1. View all devices:

Basically view the database as is, formatted for readability

1. Check-out Process:

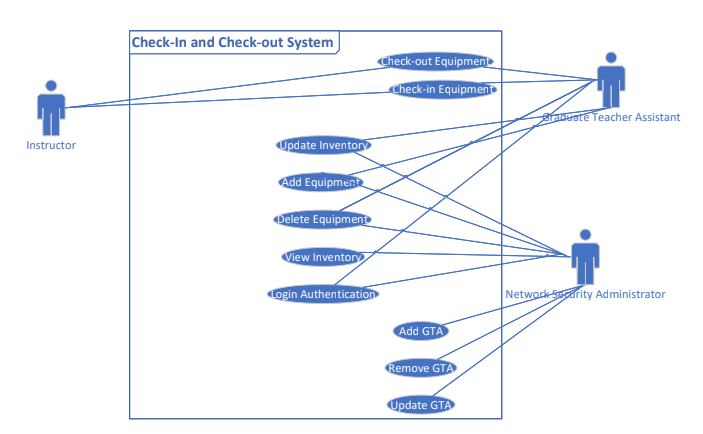
Fill in the following fields: name, email, panther\_ID, signature, device\_ID, checkout\_date, assigned\_by saved to the database.

1. Check-in Process:
2. Lookup a specific record by device ID

Fill in the following fields: device\_condition, checkin\_date

1. Generate Report of all:
   1. Checked Out Devices (If the device ID is found in the database)
   2. Overdue Devices (When Today’s Date > due date)
   3. Damaged/Broken/Lost Devices (When the condition is either damaged, broken, lost)

**Use-Case Diagram**

****

**Requirement Specifications**

**Use Case**: View Inventory Database (interchangeable)

**Summary**: Admin or TA requests to view the inventory database for raspberry pi’s

**Basic Course of Events**:

1. TA or Network admin requests to view inventory
2. They will be prompted to authenticate themselves
3. Once validated, an inventory report will be generated and displayed

**Alternate Paths**: At Step 2, the user can cancel the request for an inventory report

**Exception Paths**: At Step 3, if the user’s information cannot be validated, then they will not be permitted to view the inventory

**Precondition**: The Network Admin must have completed use-case “Login Authentication”

**Postcondition**: The Network Admin can view the equipment within the inventory database

**Use Case:** Check Out Equipment

**Summary:** GTA scans Item to check out

**Basic Course of Events:**

1. GTA Scans the barcode on the Equipment box
2. The information is entered into the database
3. The GTA completes use-case “Update Inventory”

**Alternate Paths:** If the GTA finds any missing equipment from the container they can cancel and find another container.

**Exception Paths:** If the student’s credentials are incorrect they are not allowed to check out equipment.

**Preconditions:** The Login Authentication is completed successfully

**Postconditions:** The equipment is given to the student and is no longer available to be checked out again.

**Use Case:** Check-In Equipment

**Summary:** GTA can check equipment back into the database system

**Basic Course of Events:**

1. GTA checks to make sure all the equipment is within the container
2. Then they will proceed to scan the barcode on the box
3. Information is stored into the database
4. The GTA completes use-case “Update Inventory”

**Alternate Paths:** If the GTA finds any missing equipment from the container they can mark it “replaced” and add reasons why

**Exception Paths:** need to think for later

**Preconditions:** Student who checked equipment out should check it back in with their credentials

**Postconditions:** The equipment is given to the GTA and is now available to be checked out again.

**Use Case:** Update Inventory

**Summary:** Admin or GTA can make an update to the inventory of equipment

**Basic Course of Events:**

**Alternate Paths:**

**Exception Paths:**

**Preconditions:**

**Postconditions:**

**Use Case:** Add Equipment

**Summary:** The Network Admin has the permissions to add more equipment to the database

**Basic Course of Events:**

1. The Network Admin requests to add new equipment to the database
2. They will be prompted to authenticate themselves
3. They can then enter the serial identification that corresponds to the barcode that belongs to the equipment box

**Alternate Paths:** At Step 2, the user can cancel the request for adding new equipment

**Exception Paths:** At Step 3, if the user’s information cannot be validated, then they will not be permitted to add to the inventory and will be displayed an error message

**Preconditions:** The Login Authentication is completed successfully

**Postconditions:** The Network Admin can add more equipment to the inventory database

**Use Case:** Delete Equipment

**Summary:** The Network Admin has the permissions to delete equipment from the database

**Basic Course of Events:**

1. The Network Admin requests to delete equipment from the inventory database
2. They will be prompted to authenticate themselves
3. They can then enter the serial identification that corresponds to the barcode that belongs to the equipment box

**Alternate Paths:** At Step 2, the user can cancel the request for deleting equipment

**Exception Paths:** At Step 3, if the user’s information cannot be validated, then they will not be permitted to delete from the inventory and will be displayed an error message

**Preconditions:** The Network Admin must have completed use-case “Login Authentication”

**Postconditions:** The Network Admin can delete one or more equipment from the inventory database

**Use Case:** Add Graduate Teaching Assistant (GTA) User

**Summary:** The Network Admin can add another GTA user to the system to handle equipment inventory

**Basic Course of Events:**

1. The Network Admin requests to add a GTA user to the user database
2. They will be prompted to authenticate themselves
3. They can then enter the name of the GTA into the user database
4. The user database will be updated

**Alternate Paths:** At Step 2, the user can cancel the request for adding new GTA users

**Exception Paths:** At Step 3, if the user’s information cannot be validated, then they will not be permitted to add a user and will be displayed an error message

**Preconditions:** The Network Admin must have completed use-case “Login Authentication”

**Postconditions:** The Network Admin can add one or more users to the user database

**Use Case:** Remove Graduate Teaching Assistant from the user database

**Summary:** The Network Admin can remove a GTA user from the system for handling equipment inventory

**Basic Course of Events:**

1. The Network Admin requests to remove a GTA user from the user database
2. They will be prompted to authenticate themselves
3. They can then enter the name of the GTA into the user database
4. The user name will be removed and the user database will be updated

**Alternate Paths:** At Step 2, the user can cancel the request for deleting GTA users

**Exception Paths:** At Step 3, if the Network Admin’s information cannot be validated, then they will not be permitted to remove a user and will be displayed an error message

**Preconditions:** The Network Admin must have completed use-case “Login Authentication”

**Postconditions:** The Network Admin can remove one or more users to the user database

**Use Case:** Update Graduate Teaching Assistant

**Summary:** The Network Admin can update a GTA user’s name in the system for handling equipment inventory

**Basic Course of Events:**

1. The Network Admin requests to update a GTA user name within the user database.
2. They will be prompted to authenticate themselves
3. They can then enter the current name of the GTA into the user database
4. The NA is requested to change the user’s name to another name via a display
5. The previous user’s name will be removed and the new user’s name will be updated into the user database

**Alternate Paths:** At Step 2, the user can cancel the request for updating the GTA user name

**Exception Paths:** At Step 3, if the Network Admin’s information cannot be validated, then they will not be permitted to remove a user and will be displayed an error message. They also can’t change the user’s name to another current user’s name

**Preconditions:** The Network Admin must have completed use-case “Login Authentication”. The new name (first, last) cannot already be within the system

**Postconditions:** The Network Admin can update one or more user names to the user database

**Use Case:** Login Authentication

**Summary:** Admin or TA are requested to log in for making edits within the inventory database for raspberry pi’s

**Basic Course of Events:**

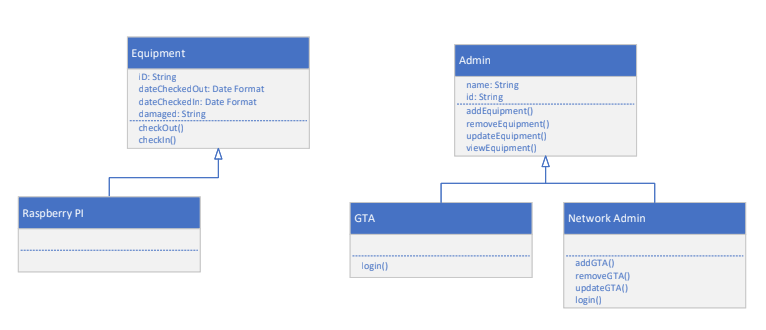
1. The user will be requested with a window with input for login username and password
2. The system will check with the user database for validation
3. The NA or TA now can access to certain properties within the inventory database system

**Alternate Paths:** The user of the application can cancel or exit out of the application at any point before logging in

**Exception Paths:** At Step 2, if the NA or GTA’s information cannot be validated, then they will not be permitted to login and will be displayed an error message.

**Preconditions:** The NA must have access to the software (don't know just yet)

**Postconditions:** The Network Admin or GTA user can have access to making edits within the equipment and user database based on privileges.

**System Modeling**

There are two major objects that will make a majority of our sub-objects. The first is the Equipment object which can be used to create different types of equipment. This includes but is not limited to Raspberry Pi’s, computers, etc… The second major object is the Admin object which can be used to create different types of administrators that will use the software. This includes Network Administrator, Graduate Teaching Assistant, and Instructor.

. An Equipment object, Admin object, and their sub-objects each come with their own attributes. An Equipment object can contain: An “ID” to identify each equipment from one another, a “checkedOut” attribute that will contain the date and time that the equipment is checked out, a “checkedIn” attribute that will contain the date and time that the equipment is checked back in, and a “damaged” attribute that will by default display “Not Damaged” and will be updated with a damage description if it is.

Each object also contains operations/actions that they can perform by default. The Equipment object contains a “checkOut()” operation that will check out that equipment, as well as a “checkIn()” operation that will check in that equipment. The Admin object also contains operations, such as: “addEquipment()” which allows Admins to add different types of equipment to the inventory, “removeEquipment()” which allows Admins to remove a particular equipment, “updateEquipment()” which allows an Admin to update a particular equipment, and “viewEquipment()” which allows an Admin to view a list of a particular equipment.

**Database Specification & Analysis:**

**Database Management System- MySQL**

***Table STUDENT:***

[

**panther\_id** (INT, PRIMARY KEY)

student\_first\_name (VARCHAR)

student\_last\_name (VARCHAR)

student\_email (VARCHAR)(\*@student.gsu.edu or \*@gsu.edu domain required)

student signature (BOOLEAN)

disclaimer\_signed (BOOLEAN)

teacher\_assistant\_id (INT, Foreign Key from teacher\_assistants(“ta\_id”))

]

***Table TEACHER\_ASSISTANTS:***

[

**ta\_id** (INT, Auto Index, PRIMARY KEY)

ta\_panther\_id (INT)

ta\_first\_name (VARCHAR)

ta\_last\_name (VARCHAR)

ta\_email (VARCHAR)(\*@student.gsu.edu or \*@gsu.edu domain required)

]

***Table KIT\_ITEMS:***

[

**kit\_id** (PRIMARY KEY)

serial (VARCHAR - ex:RPI-001)

]

***Table DEVICE\_KITS:***

[

**device\_kit\_id** (COMPOSITE PRIMARY KEY, Foreign key for kit\_items(“kit\_id”))

**item\_model** (VARCHAR, COMPOSITE PRIMARY KEY)

item\_name (VARCHAR)

item\_description (VARCHAR)

device\_condition ( ENUM: “new”, “good”, “damaged”, “broken”, “lost”)

]

***Table DEVICE\_KIT\_RENTALS:***

[

**rental\_id** (PRIMARY KEY)

checkout\_date (Today’s Date)

check\_in\_date (DATE, NULL On Default)

due\_date (DATE, estimated end of the current semester - hardcoded/changes each semester)

assigned\_ta\_id (INT, Foreign Key for TEACHER\_ASSISTANTS(assigned\_ta\_id))

device\_condition ( ENUM: “new”, “good”, “damaged”, “broken”, “lost”, Cascade on update to ***DEVICE\_KITS(***device\_condition))

]

**Appendix**

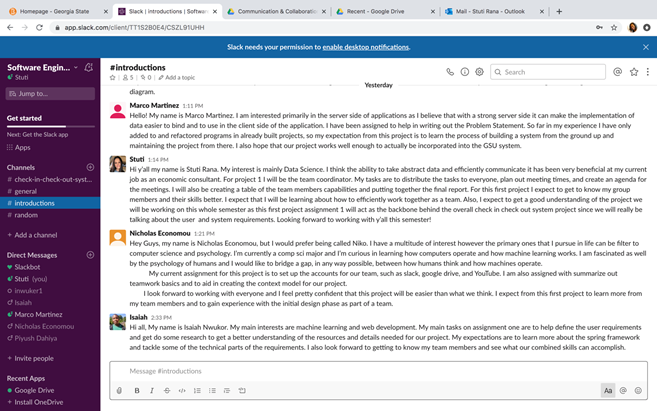
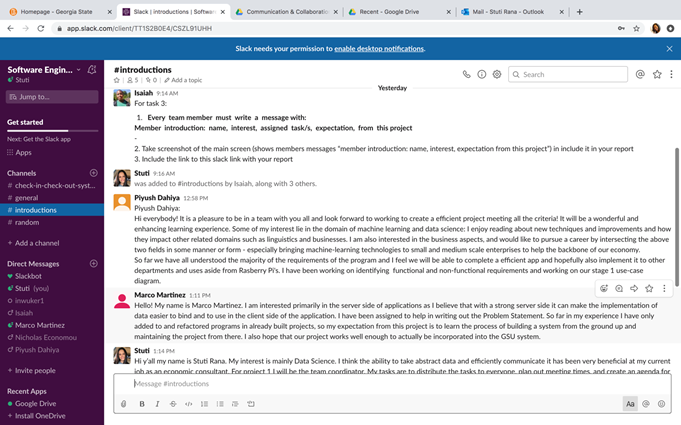
[****](https://www.youtube.com/watch?v=uqnf_qZc25k&feature=youtu.be)

* YouTube Video A2: <https://www.youtube.com/watch?v=uqnf_qZc25k>
* Previous Video A1: [https://www.youtube.com/watch?v=J6-yPZ2IZNI](https://www.youtube.com/watch?v=J6-yPZ2IZNI&feature=youtu.be)
* GitHub Link:<https://github.com/g2nimps/Raspberry-Pi-Rental-System>
* GitHub Project Link: <https://github.com/g2nimps/Raspberry-Pi-Rental-System/projects/1>
* Slack Link:<https://app.slack.com/client/TT1S2B0E4/CSQCDAX27>
  + Slack domain: softwareengin-o1q8410.slack.com

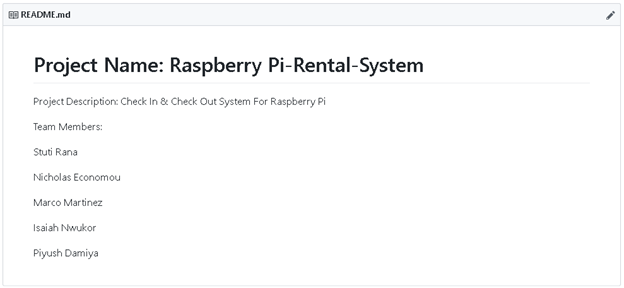
**Resumes:**

* Nicholas Economou: [Resume](https://drive.google.com/open?id=1GzNt60idK6bRM76N0J4_02I-dLnjCvLd)
* Isaiah Nwukor: [Resume](https://drive.google.com/open?id=1JSkY7ReHRVzCGCRzbW0n9ciDaOLnE0V3)
* Marco Martinez: [Resume](https://drive.google.com/open?id=1J1o5ND7PHBTW4hJENE0LxFppSf_S1CUJ)
* Piyush Dahiya: [Resume](https://drive.google.com/open?id=1fufgQkbUpHDOCy3KNkkPRNtaN5D2zHTc)
* Stuti Rana: Resume

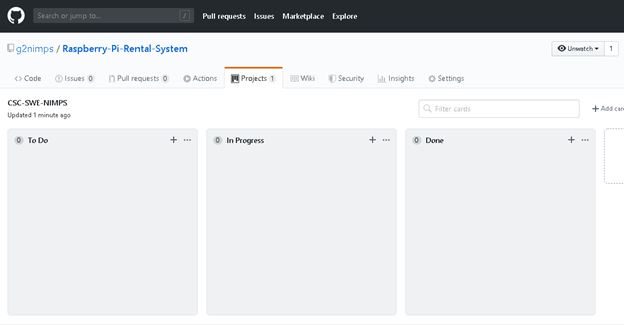
**Slack Screenshots:**

****

**GitHub Screenshots:**



Initial Project Board Creation



Project Board - Assignment 2:

