April 1st, 2018
ITCS 4155
Group 11

RFIDefense

Jake Mecimore, Justin Farkas, Fiacre Indagiye, Young Jung, Charles Valdez

1. Project Overview

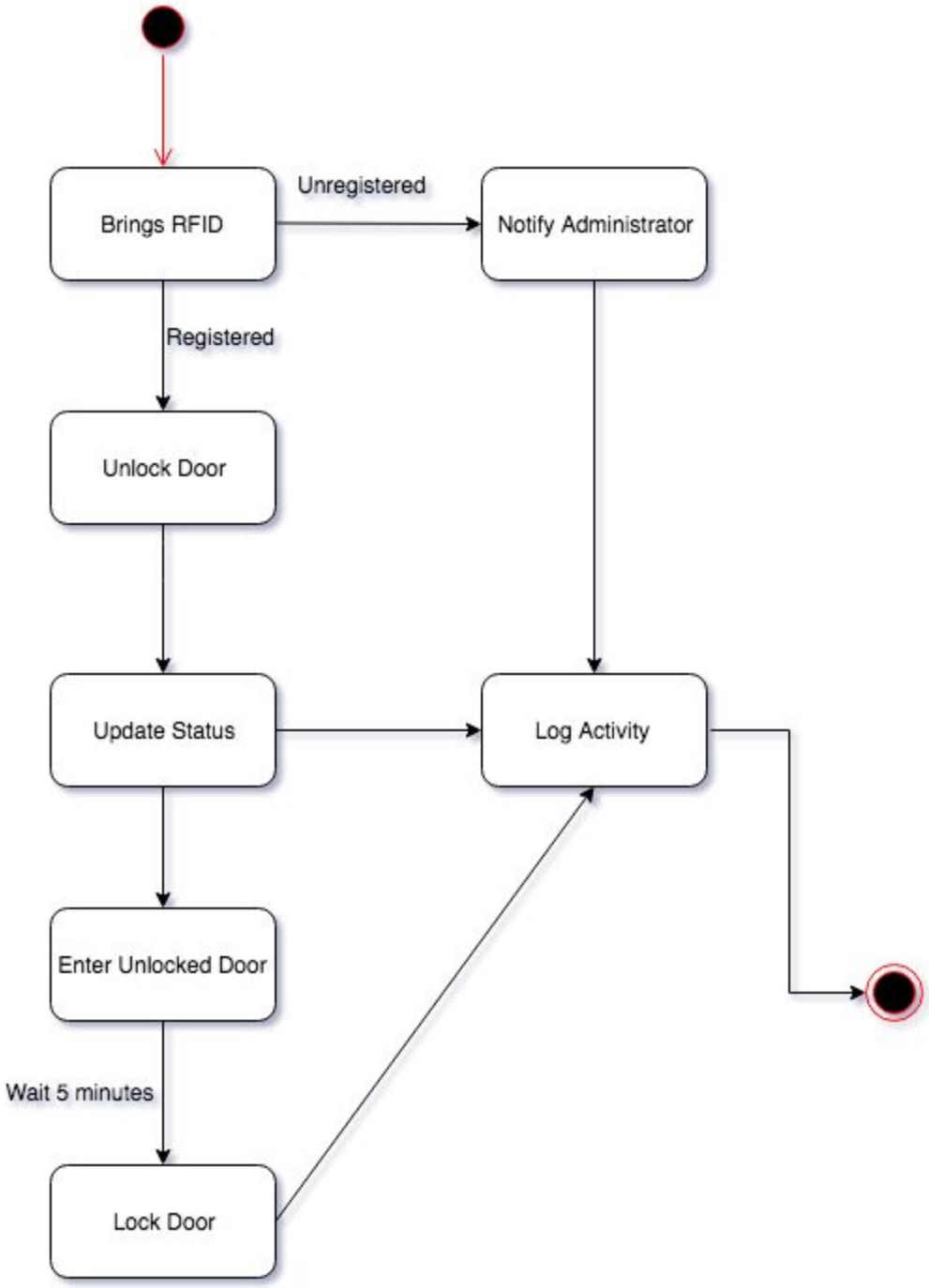
RFIDefense is a home monitoring/security system that simplifies the process of keeping your home and everything in it secure and protected. The purpose of this product is to take the conventional method of having a mechanical lock and key for every door in your house and get rid of it completely. The RFIDefense system uses RFID technology and a Raspberry Pi to replace the old lock and key system. In switching to a technology-based system, we can offer users the ability to connect to the system and control it while away from home using our RFIDefense HOME mobile app.

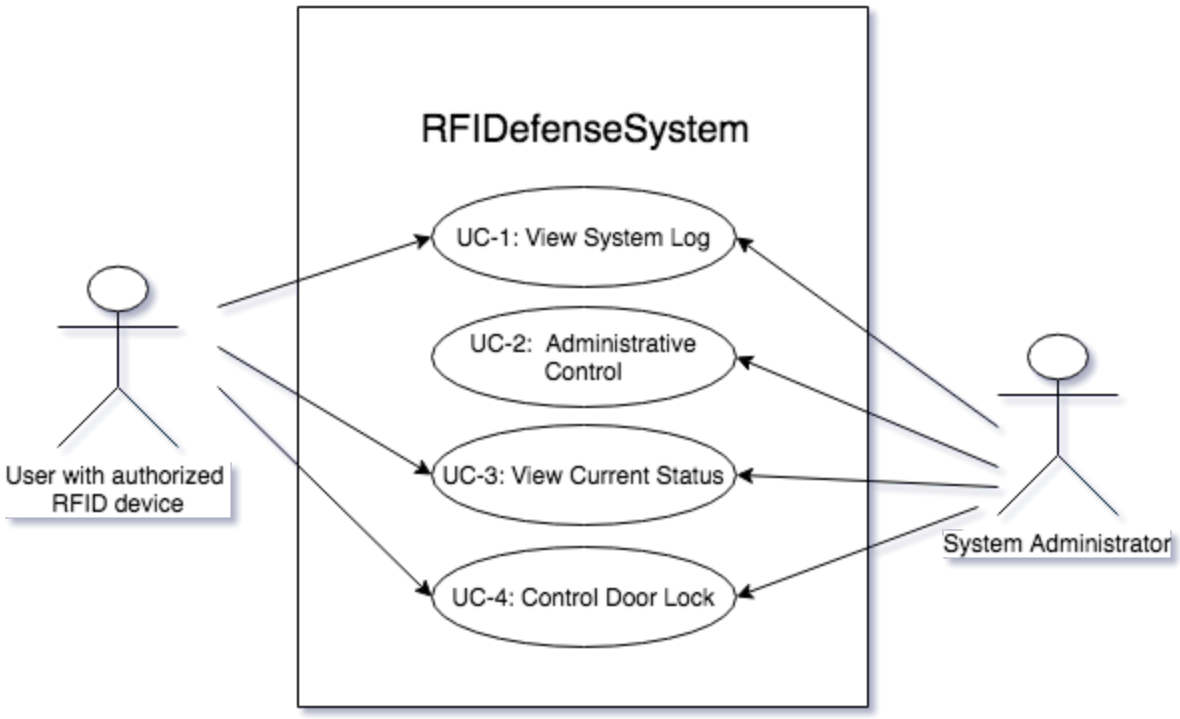
One of our biggest use cases for the system is families with children. With our product, parents could monitor the system to see if their children have either come home and entered the house or have left the house. Beyond that, parents could lock and unlock the door remotely for their children if they were to forget their own access key. Our system would give parents a way to ensure their kids are where they are supposed to be when they are supposed to be there from their smartphones.

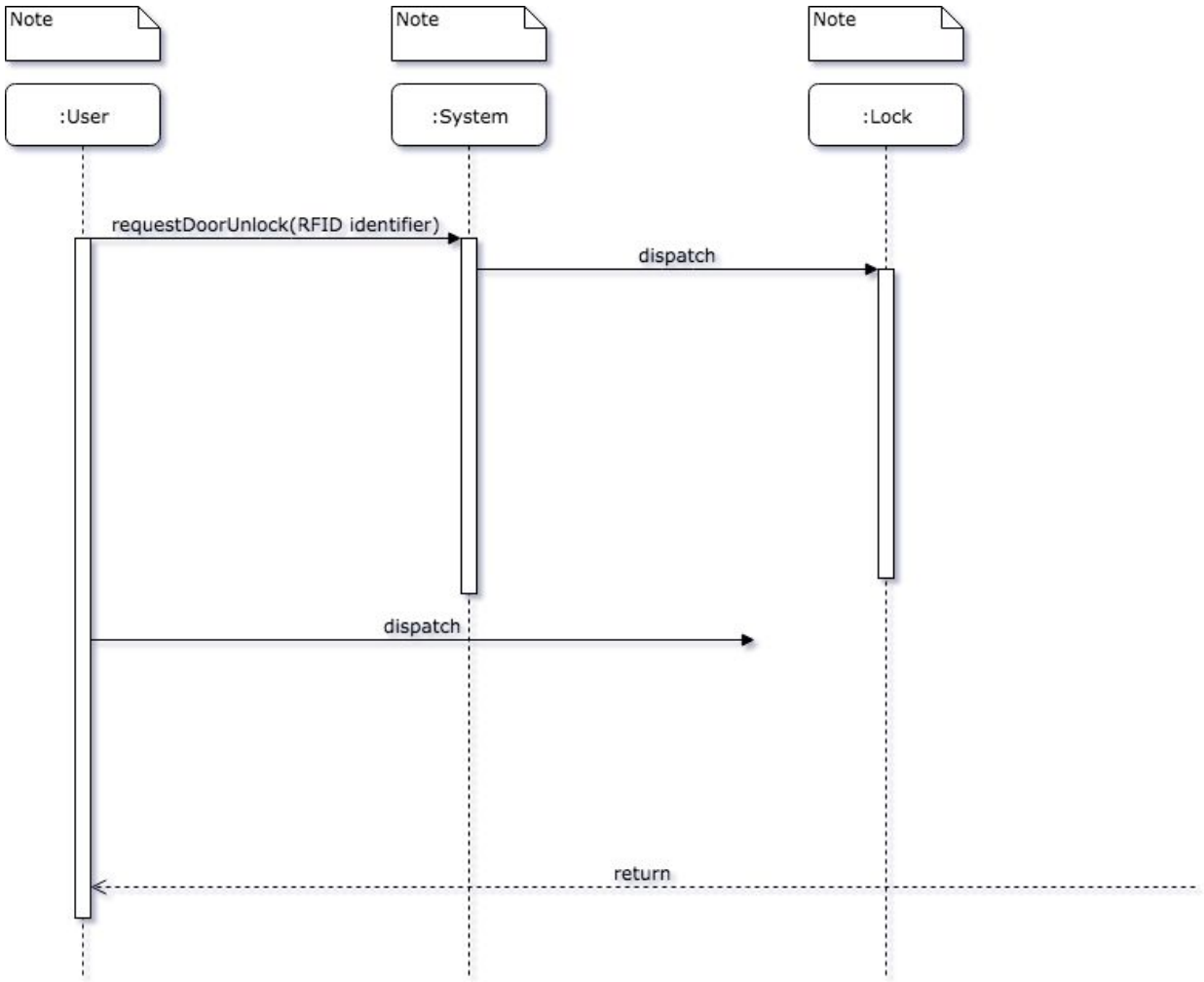
2. **Architectural Overview**

RFIDefense is designed to be used in any residential or commercial building to create a hassle free locking system. Our security system will be used by families or employees using registered RFID key/tags that will permit access once in contact or near our RFID sensors. The systems Admin or Parental figure of the household will be able to keep track of the ID's used through our mobile application. For our locking mechanism, we created a python script that runs on a raspberry pi that will take in RFID string values and compares it to the information in our database. For our database we implemented a web server using SQL to take in user id, password, and RFID associated with the id and passwords. Lastly for our Mobile app we are creating the app on the android platform using Java for backend development and XML to incorporate the user interface.

a. Subsystem Architecture

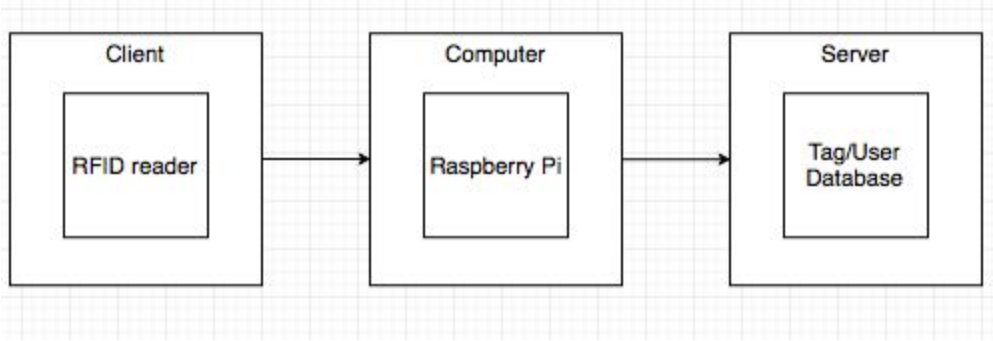



RFIDefenseSystem
UC-1: View System Log
UC-2: Administrative Control
UC-3: View Current Status
UC-4: Control Door Lock
User with authorized RFID device
System Administrator



Our projects consists of 3 subsystems.Tthe user, the database/system, and the lock. The User is represented by a unique RFID that allows access to the functions of the system. The Database contains the users ID, password, and specific RFID linked with the user. The Lock will receive RFID tags and check the database to determine if the RFID is registered and provide access or deny the user.

b. Deployment Architecture



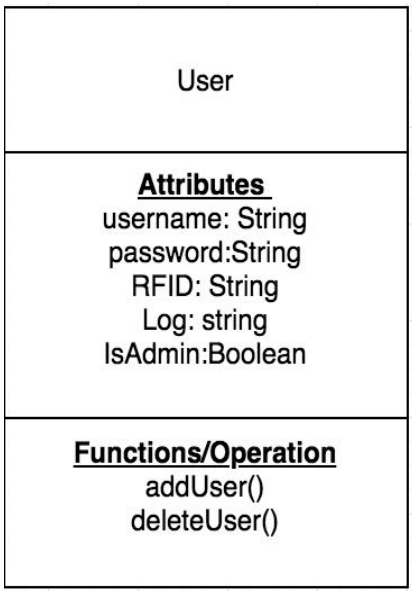
3. Persistent Data Storage

There will be a couple of different types of data that we are going to be storing in our application. All of our data will be stored on a SQL database that will live on a web server that our mobile application is hosted on. The first table will be a table that has usernames and passwords that are set up for the administrator who will have access to the app. Also in this first table is a column that will store the connection to the Raspberry Pi device that will be the central hub for the security system. Each record in the main table will have its own table that stores the user data. The table will store the RFID tags along with the user status associated with the tag. As RFID devices are added to the system by the admin, they are added to the table and assigned a user status status. If needed, admins are able to edit rows to change user status. Admins are also able to remove rows from the table using the RFID tag as a key.

4. Global Control Flow

RFIDefense will be an event-driven system. The majority of the system deals with the unlocking and locking of doors and that is controlled by approved RFID tags. The event that drives all of our system is the scanning of a tag. The sensor sends the tag to the Raspberry Pi and then the program checks if the tag is registered with the system When the status is returned, the system reacts to whether or not the tag has access. If the tag doesn't have access, the door remains locked and an unauthorized attempt is added to the activity log. On unauthorized attempts the record in the activity log is also sent to the mobile application user in the form of a push notification. If a tag that has access is scanned, then the system tells the Raspberry Pi that the door can be unlocked and sends a signal to the lock to unlock. The authorized access is added to the activity log. The system waits 10 seconds and then sends another request to the lock to lock the door back up.

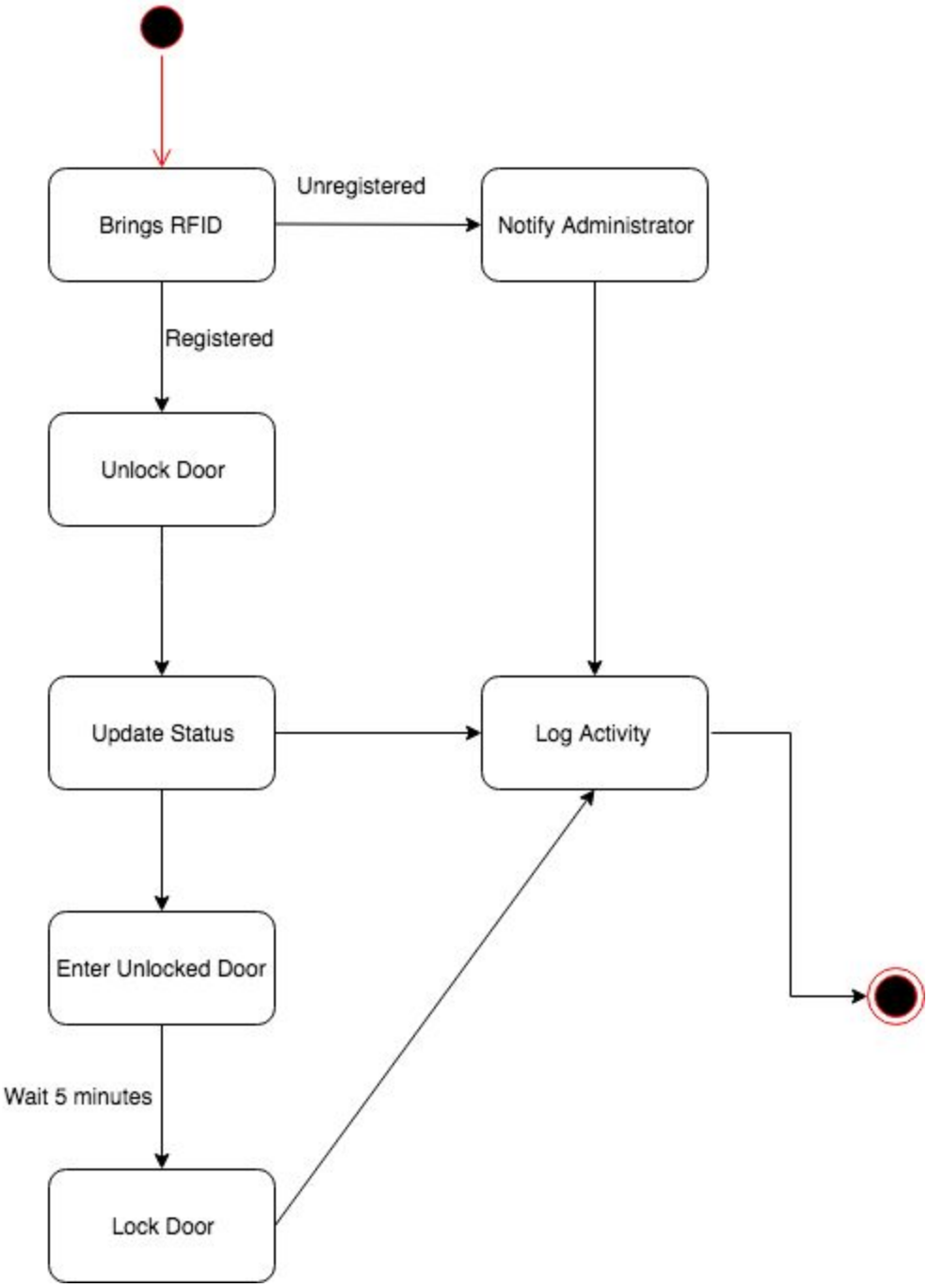
5. **Static View**



Our user table holds the users username, password, and RFID. With these string values we are able to create a Log that the user will be able to pull. Our user table will also check to see if admin privileges are available to create and delete new users as seen fit.

6. Dynamic View

The system’s behavior UML diagram



As the diagram shows, whenever a new RFID is used the system checks if the RFID is registered. If the RFID is already registered, the door unlocks and the status on the app also changes to display the new status of the door on the app. If the RFID is not registered, the door remains locked and the admin is notified of the suspicious activity. After both actitions a log is kept. If the RFID is recognized, the door is unlocked, after 5 minutes the door automatically locks again and that activity is also logged. The administrator has access to all the logs therefore he can see whether or not the door is locked or unlocked by checking the logs.