Project Goals:

At the start of the semester, our goal was to build a portable, non-intrusive smart unlocking system for dormitory rooms that combined both hardware and software components to enable secure keyless access. We planned to use facial and voice recognition through a Flask backend, which, if authenticated, would send an unlock signal to a Bluetooth-connected ESP32 microcontroller, triggering a stepper motor to physically turn the lock using a flexible steel cable.

We've made strong progress toward these goals, especially on the software and integration side, and have successfully created a mock demonstration of the unlocking system using a breadboard-based prototype. The facial and voice authentication pipeline is functional, and the Android app reliably sends unlock commands to the ESP32 over WiFi (changed from bluetooth to WiFi connections between our Flask server and the ESP32), which in turn triggers the motor as expected. The mechanical setup remains portable and satisfies the requirement of being mountable without modifying the dorm door.

While the custom PCB is not yet operational, we hope to have it ready for the final demonstration. In the meantime, the breadboard implementation has allowed us to validate all key subsystems, including power regulation, communication, and actuation.

In summary:

- Software goals were largely met: backend, app, and authentication pipeline work reliably.
- Hardware goals are partially met: the system works as expected on a breadboard, and final integration on a PCB is in progress.

Expectations:

Our expectations, as laid out in the team contract, were mostly met throughout the semester. Each team member came prepared to meetings and was consistent with progress and each member contributed meaningfully to discussions about the project and with the design of the project. Each groupmate communicated openly about any delays due to exams or other commitments and we were able to navigate delays well. We were responsive on messages and transparent about individual workloads. This helped maintain momentum even during crunch times.

The main area where we occasionally fell short was meeting twice a week outside of TA meetings, as originally planned. Due to very busy schedules, particularly around exam

weeks and major deadlines—we sometimes had to scale back to just one meeting. However, we made up for missed sessions by making sure we were messaging about the project and still trying to move forward.

Overall, we believe that the group contract was upheld pretty well. We respected each other's time and communicated effectively.

Roles:

At the start of the course, we gave everyone general roles based on what they were good at or interested in. As the semester went on, we split up parts of the project to make progress faster, but we still worked together and helped each other out when someone got stuck.

- Raghav worked on most of the coding for the app and the Flask server that handles authentication and sends unlock commands.
- Yuhao and Arnav focused on the PCB design, trying to move our breadboard setup to a more permanent circuit board.
- Arnav also took the lead on writing the ESP32 code that connects over WiFi and controls the motor.
- When it came to building circuits and soldering, all of us worked together in the lab to get everything wired up and tested.

We didn't assign an official leader—everyone took charge of their part, and we checked in with each other often to make sure everything was working together.

Agenda: How did your team make decisions about the project? How were goals set? When an issue with the project came up, how did your team plan to fix it?

Our team made decisions about the project using a rational process that factored in time. For example, decisions related to parts ordering tended to be quite conservative. This meant we ordered multiple copies of parts and also ordered different iterations of chips since it was not always entirely clear what would work in practice. There was a tradeoff here between getting the project done, and cost, and we chose to prioritize finishing in the allotted time.

Goals were set on weekly bases. We wanted to have something tangible to show to our TA each week. Deliverables included app UI features, PCB designs, PCB/parts orders, breadboard

demos, etc. Goals were not always met due to other obligations but they provided a guiding point for us to work towards.

Issues arose in this project. An example of certain issues was that our first order of parts disappeared, and we are currently not sure if they got stolen. As a result, we had to rapidly order a new set of parts. However, we also asked ourselves if anything else could be included in the order to save time. Additionally, for the breadboard demo, we were working with a faulty ESP-32 chip, so we obtained a new one from another group temporarily to get a demo working. Our general approach to solving issues would be to ask “What can we do given how much time we have?” Often the solutions were not complete, but we attempted to solve as much of the problem as we could.

Team Issues: This section should cover team-related issues that your group encountered during the course. What sort of problems did you run into? How were they dealt with? Was the process set out in the team contract followed? In hindsight could you have done things differently to have a better team experience?

We encountered issues with component mismatches during the project. To address this, we ordered at least four units of each component—two through-hole and two surface-mount versions—to ensure compatibility with both breadboard testing and PCB implementation.Additionally, we found that some capacitors and inductors did not meet the required values, so we used series and parallel configurations to achieve the desired specifications.

One of the biggest issues we ran into was having some of our parts stolen from the lab, which set us back in terms of both time and budget. We had to reorder several components, which delayed our progress and forced us to rely on backup parts for a while. This delayed our PCB soldering process by a lot. We also had a faulty ESP32 dev board that wasn’t obvious at first. It took a lot of time to realize the board itself was the problem, not our code or wiring of the breadboard. This also wasted time, but we eventually identified the issue and replaced the board..