

## First Iteration Demo

### Demo Date:

Wednesday, November 7, 2018 at 7:00 PM

### Github Repository:

<https://github.com/TeamTechnica/Skybot>

### Challenges that arose during demo:

None. However, Ashna, our Project Mentor, gave us tips on how to improve our pre-commit configuration to allow developers to easily check in "in progress" branches.

### Project Revisions

In our initial proposal, we wanted to create a Facebook chatbot. However, due to recently increased security and privacy measures on Facebook, we were unable to create a chatbot using Facebook. We decided then to utilize Twilio instead to create an SMS bot. The following features were revised:

- Users will communicate with Skybot via SMS using a mobile device.
- Users will be verified using a Columbia provided UNI through email verification.
- Matched users will be matched via email.

For the first iteration Skybot, our team's goal was to set up our environment and develop the infrastructure needed to create an SMS bot. This involved:

- Setting up a Github repository
- Implementing continuous integration using Travis CI
- Integrating the Twilio API for SMS functionality
- Integrating the SendGrid API for email verification
- Creating a SQLite Database for user data storage
- Using Heroku Webhooks to handle requests

## User Stories

At this stage in the development process, we were able to demo the following user story:

As a rider that is interested in creating a new Skybot user account. My criteria for satisfaction is that I receive a welcome message from Skybot when I first message the bot and to be prompted to enter my user information (UNI) in order to verify that I entered correct contact information and that I am a Columbia University student.

During the demo, Ashna was able to message the bot, provide her UNI, and receive a verification in her Columbia email with a verification code. Her number was added to the database and is stored as a user.

## Continuous Integration & Technology

For our pre-commit, we used a publically available pre-commit configuration that was provided on Github. Before code is pushed to master, it must pass all of the lint and style checks. For our post-commit, we used Travis CI to run tests developed with the Python unit testing package. Before being able to commit to master, developers also have to invite at least one team member to review their code before merging to master.

Technology stack:

- Development Framework: Python (Flask)
- Unit Testing Tool: PyUnit, Nose Tests
- Static Analysis Tool: Pylint
- Continuous Integration Tool: Travis CI
- Data Storage: SQLite3, SQLAlchemy (for queries)
- Version Control: Github
- APIs: Twilio, SendGrid, Lyft API