FLYERSCAN

A REPORT SUBMITTED AS A REQUIREMENT FOR CSE 155: INTRO TO HUMAN-COMPUTER INTERACTION

Frank Olotu
Role: Design & Development (Interface & API)
University of California, Merced

Johnny Ly
Role: Design & Development (Interface)
University of California, Merced

Gabriel Navata
Role: Development (API)
University of California, Merced

Austin Myhre
Role: Design & Development (Interface)
University of California, Merced

May 5, 2023

ABSTRACT

FlyerScan is an app designed to help students keep track of on-campus events by scanning event flyers and extracting event details into a personal calendar. Using Optical Character Recognition (OCR) and Named Entity Recognition (NER), the app identifies parameters pertaining to a calendar event on physical flyers and imports the data into the user's personal calendar app. The user can confirm and edit any necessary changes using the app's speech detection and keyboard text input.

1 Goals and Motivations

Tracking campus events can be difficult and time-consuming. Physical flyers often clutter bulletin boards and can be hard to find. Manual entry of event details into a personal calendar is prone to errors. With that in mind, the primary goal of FlyerScan is to simplify the process of tracking on-campus events for students. By automating the extraction of event details from flyers, students can easily add events to their personal calendars and stay informed about upcoming activities. The project aims to satisfy the following criteria:

- Accounts for users' needs and requirements
- Requires software development
- Uses a machine learning or recognition component
- Builds interaction techniques

2 Requirements and Prioritization

The requirements for FlyerScan are categorized using the MoSCoW method, which consists of Must have, Should have, Could have, and Won't have categories.

2.1 Must have

- Able to scan a flyer and determine it is a flyer
- Able to extract date, time, location, and event from the flyer
- Generate calendar file from the information pulled from the flyer
- Review screen with final event information and option for manual correction with keyboard
- History of past scans

2.2 Should have

- Edit with voice command
- Share event information with friends

2.3 Could have

- Reminder function for 2 days in advance before event
- Machine learning aspect to learn from mistakes made in past scans

2.4 Won't have

- Will not work on things that don't contain information pertaining to events

3 Design Process

The design process for FlyerScan involves understanding user needs, defining the user interface, and refining the interaction experience.

3.1 Understanding User Needs

To ensure the app meets the needs of its target users, the team conducts interviews, surveys, and usability tests with students. This information helps identify the most important features and requirements for the app. Additionally, the team gathers feedback on the app's performance, ease of use, and potential pain points to address in future iterations.

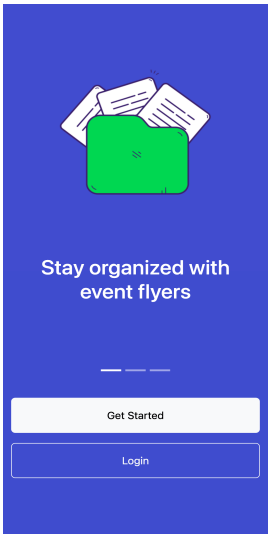
3.2 User Interface Design

The user interface is designed to be simple and intuitive, allowing users to easily scan flyers, review extracted event details, and edit the information as necessary. The interface incorporates visual and auditory feedback to enhance user interactions. With a mobile responsive design, FlyerScan works on various screen sizes ranging from smartphones to laptops to help make the app accessible to multiple users. The app's user interface was intentionally designed for easy one-handed use, with UI elements strategically placed toward the bottom of the mobile screen to minimize thumb movement and ensure a more seamless experience.

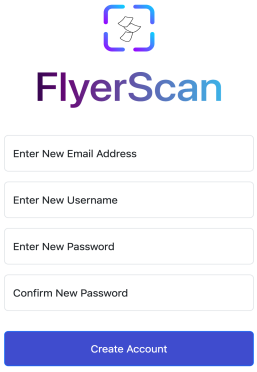
3.3 Interaction Design

The interaction design focuses on providing a seamless experience between the user, the app, and the extracted event information. This includes the ability to edit event details using voice commands or keyboard input and the option to share event information with friends.

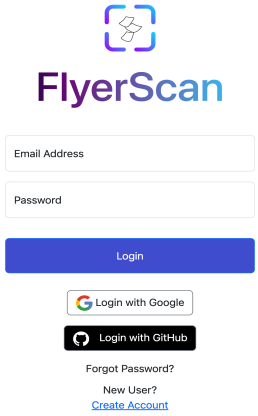
4 App Screenshots



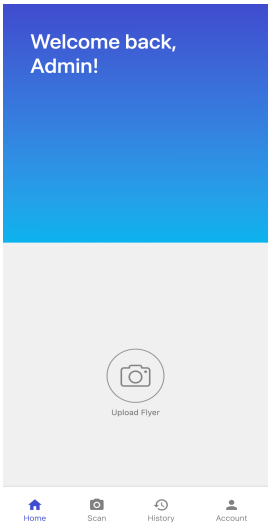
(a) Landing Page



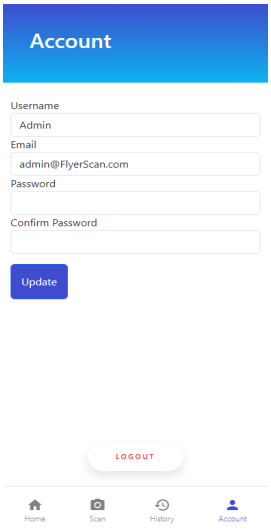
(b) Registration Page



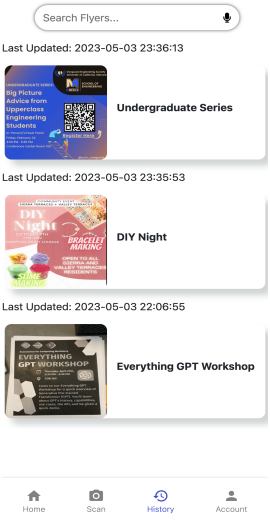
(c) Login Page



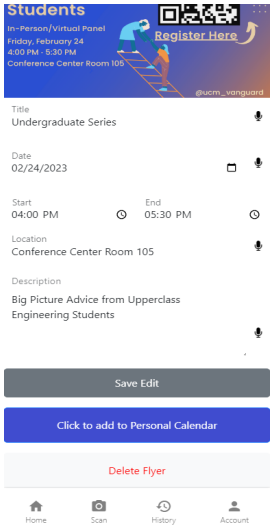
(d) Home Page



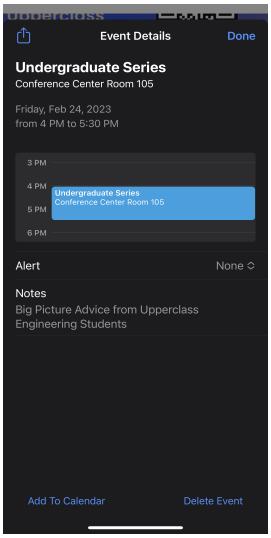
(e) Accounts Page



(f) History Page



(g) Edit Page



(h) Add to Calendar Page

Figure 1: FlyerScan App Screenshots

5 Implementation

FlyerScan's implementation consists of two main components: hardware and software.

5.1 Hardware

The hardware required for FlyerScan includes a smartphone or device with a built-in camera and microphone, which enables users to scan flyers and provide voice commands for editing event details. With the app's user interface built with a mobile-first design, the responsive UI works from smaller screen sizes like smartphones to larger devices like tablets, laptops, and desktops.

5.2 Software

The software stack for FlyerScan includes Python, Flask, JavaScript, HTML, CSS, OCR (Optical Character Recognition), NLTK (Natural Language Tool-Kit), and BERT (Bidirectional Encoder Representations from Transformers). The app uses OCR to extract text from scanned flyers and natural language processing (NLP) techniques with the help of BERT to identify event details. The Flask framework is used for building the backend API, while the frontend is developed using JavaScript, HTML, and CSS. In addition to the core software components, FlyerScan leverages cloud-based services for scalability and data storage. This enables the app to handle a large number of users and efficiently store their scan history and event details. Furthermore, the use of cloud services allows for seamless updates and improvements to the app without causing any disruption to the user experience.

6 An Example Use Case

A student comes across a flyer for an upcoming guest lecture on campus. They open the FlyerScan app, point their smartphone camera at the flyer, and the app scans the flyer, extracting the event's date, time, location, and title. The student can review the extracted information, make any necessary edits using voice commands or keyboard input, and add the event to their personal calendar. They can also share the event details with friends who may be interested in attending.

7 User Feedback

Upon completing the initial implementation of FlyerScan, the team conducts usability tests with students to gather feedback on the app's performance, ease of use, and overall usefulness. This feedback is invaluable in identifying areas for improvement and refining the app's features and functionality.

8 Plans for the Future

As we continue to develop FlyerScan, we plan to focus on several key areas for improvement and feature enhancements. These enhancements aim to make the app more robust, user friendly, and capable of handling a wider array of event flyers.

- **Improved OCR and NLP:** We plan to invest in improving the OCR and NLP capabilities of the app to better accurately extract event details from various flyer designs and layouts. This will include fine tuning the BERT model we use to better recognize parametric entities (date, time, title, locaton), we will also begin training a separate model to infer from mistakes made from choosing the wrong parametric entity by learning from the users edits to the field parameters, we plan on doing these by exploring state-of-the-art techniques in OCR and NLP.
- **Personalized Event Recommendations:** In the future, we aim to add a personalized event recommendation system that can suggest events based on user interests and past attendance. This will help users discover relevant events and further enhance the overall user experience.
- **Crowdsourced Event Database:** In order to improve the accuracy and reliability of event information, we plan to create a crowdsourced database of campus events. Users can contribute to this database by submitting event details and verifying the accuracy of existing events. This will help to create a comprehensive and up-to-date source of event information for all users.
- **User Interface Enhancements:** Based on user feedback and ongoing usability testing, we will continue to refine the user interface to make it more intuitive, visually appealing, and responsive to different devices and screen sizes.

9 Conclusion

FlyerScan aims to provide a simple and convenient solution for students to keep track of on-campus events by scanning event flyers and automatically extracting event details into their personal calendars. By incorporating user-centered design principles and leveraging machine learning and natural language processing techniques, FlyerScan offers a streamlined event tracking experience for students. With continued development and refinement, FlyerScan has the potential to become an essential tool for students to stay informed and engaged in campus life.