Department of Computer Science & Engineering
The University of Texas at Arlington

Project Charter
CSE 4316: Senior Design I
Fall 2018

WeDiscover
iSpy

Abdul Mannan
Cameron Howard
Gaganjeet Singh
Ivy Moore
William Truong

REVISION HISTORY

Revision	Date	Author(s)	Description
0.1	10.08.2018	AM, CH, GS, IM, WT	document creation
0.2	10.15.2018	AM, CH, GS, IM, WT	complete draft
0.3	10.20.2018	AM, CH, GS, IM, WT	release candidate 1
1.0	10.21.2018	AM, CH, GS, IM, WT	official release

Contents

List of Figures

1 Vision

The objective of this project is to provide consumers with a handy way to find products that they need locally. When users search for items online, they are usually lead to an e-commerce site, which means that users typically won't be able to receive their wanted items in the same day. Instead of waiting, with our app, consumers will be able to find out where the closest store is that carries their wanted item (using their phone's camera). If a store is a close enough, users will not have to wait for an e-commerce site to ship their item. The goal of our product is to see a future where finding everyday necessities are only a few taps away.

2 Mission

iSpy will use the camera on the user's phone to see objects in real time and give information on the object. As it recognizes the object using machine learning, it will search online for local stores that carry it. Using the user's GPS, this application will find nearby stores which still have the product in stock. Once various stores are found, the user will be notified on what stores are closet or offer the cheapest prices to buy their item.

3 Success Criteria

Upon completion of the prototype system, we expect the following success indicators from users:

- 10% faster product look-up time
- 30% gain in financial savings
- 10% gain in product delivery time

Within 3 months after the prototype delivery date, we expect the following success indicators to be observed:

- An additional 10% reduction in product look-up time
- An additional 5% gain in financial savings
- An additional 15% increase in mean time to failure (MTTF)

Within 6 months after the prototype delivery date, we expect the following success indicators to be observed:

- An additional 15% reduction in product look-up time
- An additional 5% gain in financial savings
- An additional 15% increase in mean time to failure (MTTF)

4 Background

The problem we have decided to tackle is based on a lack of information that can be known about any article of clothing. Consider the following, Jack is walking down the street and passes a stranger wearing a trendy jacket with a logo he doesn't recognize. He stops the stranger, compliments their jacket, and ask where he could find a similar one. Unfortunately for Jack, the stranger is new to town and does not know which nearby stores might carry it. The stranger thanks him for the compliment and begins to walk away, and Jack is left jacketless.

The problem in a nutshell is that there is not enough information attached to clothing to make it distinguishable. More specifically, there are plenty of attributes that can be used to identify a piece of clothing, but these attributes are not useful by themselves. For example, color, material, or type of clothing cannot identify any article of clothing by themselves. However, the aggregate of data may prove useful with other identifiers like logos that may not be well known. Using machine learning and computer vision, we can train a program to recognize these attributes to return what the clothing is or might be.

Simply identifying what the piece of clothing and displaying it to the user is not particularly useful either by itself. The user will know what the piece is, but have no way of knowing things like cost or even where it could be purchased. This can be remedied by using the GPS built into many mobile phones today to search for nearby stores, use API calls to query that item, or search online stores like Amazon for the product. Even if the exact scanned item isn't found, the results should still return clothes similar to the original product. Also, since we are aggregating information from the original clothing article and stores in the area, we could sort the results by how much the item costs, distance to nearby stores, or how much the returned image matches the original. The program will probably take multiple of these factors into account to help users find clothing they are satisfied with.

5 Related Work

There are a few products that exist that are capable of both recognizing objects and searching for them online. However, they are mostly unreliable. The Google Play Store carries an App, Google Lens [1] that despite being developed by Google, has a 3-star rating due to the unreliability of the returned results. Google Lens [3] also does other things like image searching for information, translating words, identifying plants, and more. All these additional features led to a very bloated, difficult to use application. Similarly, Bixby vision by Samsung [6] is an entire personal assistant that can search by image, translate text, and shop online. However, this app too is known for its unreliability. The problem with apps that have too much functionality without being fully tested is that they are able to do lots of things, but none of them very well. We recognize this pitfall and want to build a system that has a single purpose functionality, but is very good at it. It is also important to mention that both applications only search online, not around the location the user is at. [5]

Another product that is really similar to ours is The Iconic - Fashion Shopping application. [2] The Fashion Shopping application allows users to easily search and buy clothing, however what differentiates our application is that Fashion Shopping does not have a machine vision component.

Overall, our application will fill a very clear niche that will be useful for almost everyone, without having other extravagant features that will confuse the user or slow down our development. While our product may be similar to the ones talked about above, our product is different enough to gain traction for a potential pool of users.

6 System Overview

This system will be distributed across two major components: the mobile device, and a server. The mobile system will only need access to the server in this implementation, but the server will need internet access to provide API calls to nearby stores. The mobile device will first need to detect an object of interest either though a still image (photo) or a moving image (video). Ideally this process can happen by the user tapping a button to start its recognition. After it detects the object, the mobile device will alert the user, and pause the app temporarily with a loading message. It will then send a datagram to the server containing either the still image of the object, or an encoded version of the object. The mobile device may also send GPS coordinates if nearby stores are desired. The server will be waiting for connections as it may receive connections from multiple sources. After a connection has been made, the server will enter the processing phase. The server will be running the pretrained model that will test the incoming data against data it is already familiar with. After it has determined what kind of object it is, it can then consider other features such as logos, color, text, or material. Using these attributes, it will string together a phrase that will likely be recognized by shopping services. If the server has received GPS coordinates, it will then query specific items in that region using various API's from stores located nearby. If no stores are near or if GPS coordinates aren't sent, the application can also query purely online marketplaces such as Amazon. The server will rank each result according to cost, distance, ratings, and similarity to the original product and store a link to each product. Finally, the server will enter its post-processing phase and send a small list of items that were the most highly ranked to the mobile device along with thumbnails. Then it will terminate the connection. The list of items will become visible on the mobile device and display the thumbnail and item name to the user for convenience. The user may either close out of the menu to restart the process or select any link from the list to open it in a browser.

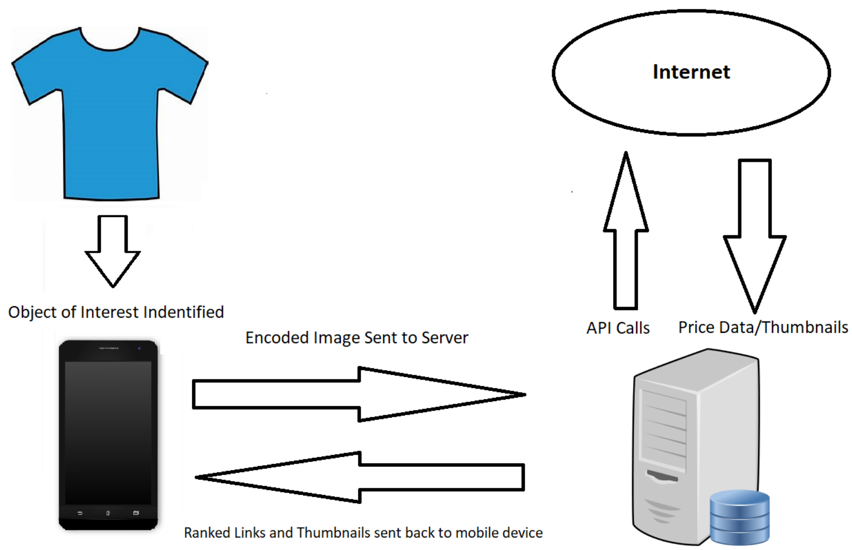

Figure 1: Example System Overview Diagram

7 Roles & Responsibilities

The stakeholders of the project will be the five people who worked on this project throughout the year. Since there are no sponsors for this project, there is no one else invested in the success of the program. As the product is developed, any customer feedback should be given to Ivy Moore as a point of contact.

As a group, we decided to split up the tasks according to the total amount of work we believe is required for that specific task. For our project, one person will be designated to work on the machine learning side of the project (Gaganjeet Singh). They will handle what the server will receive, how the information will be encoded, how new information will be matched against the pretrained model, and the attributes that will be generated for the search. Another person will be responsible for the computer vision of the program (Abdul Mannan). This person will work to make a mobile device application that will be capable of identifying objects of interest in a photo/livestream. A designated floater will go between both groups and help them during different phases of the project as each member will likely have higher workload at different times of the project (Cameron Howard). This sub-group will also need to be able to send datagrams from the mobile device to the server, so they will work together to interface both parts of the system.

Finally, the last two people will be designated to work on the various API calls that will need to be implemented for the search functionality (Ivy Moore, William Truong). The reason we decided to put two on this project is because the amount of research that will need to go into searching for the best product. This includes researching which stores to use, looking through each stores API documentation, acquiring keys, running tests to see which phrases return the best results, searching APIâs by coordinates, and merging results that have been obtained through this process. Then they will need to sort the data by location, cost, potential match to rank them by usefulness. They will then be sent back to the user.

We will maintain the same Product Owner throughout the project because there is little reason to change it. The Scrum Master will also stay the same throughout the project (Ivy Moore). The roles that might change are probably going to be the technical roles as we progress through the project and have a better understanding of how much work might be needed to complete a task.

8 Cost Proposal

The major costs for the application will result from cloud hosting, given the team decides to host the application on Amazon Web Services or a similar service. There are some major benefits of choosing to host the application on the cloud. Cloud shifts the costs of owning and maintaining hardware configuration. Seamless data access, reliable storage services, security, affordable pricing because of competition among big cloud computing companies, and flexibility and scalability are some of the major justifications for using a cloud platform. This project requires unhindered communication among team members. Although the team is not completely sure at this point, we might need some subscriptions for team collaboration. We will be using OpenCV for processing images, and some of the feature detection algorithms are not free for commercial use. The team still has to determine if we need to implement a fast, patented feature detector like SURF (Speeded Up Robust Features) algorithm, because there are other feature detectors available which do not require any licence for commercial use.

8.1 Preliminary Budget

Module	**Estimated monthly budget**	**Six month total**
Cloud computing and storage	$30.00	$180.00
Communication/Collaboration tool subscription	$8.00	$48.00
Web page with link to the application	$2.00	$12.00
Google Play store one time cost	$25.00	$25.00
Patented algorithm license	$??	$??

Table 1: High level budget table for components, fabrication, licenses, hardware etc.

8.2 Current & Pending Support

So far there is only one funding source for the project: The CSE Department. The maximum budget is about $800.00. It is unexpected that there will be another funding source.

9 Facilities & Equipment

We will be using a lot of technologies and a multitude of resources for completion of iSpy application. The primary physical location for team meetings is the CSE senior design lab. The team might also setup meetings at other convenient locations, accommodating any change of plans. We might need a centralized android device that has a processor powerful enough for image processing. However, this might not be a priority because Android Studio allows to run the application on virtual android devices.

However, if the team decides to build a local server, we might have to build a powerful machine with high-end graphical processing units that can process images and generate responses close to real-time. We can decide where this server will be placed but then we have to take the power consumption into account. We can utilize the allocated budget provided by the CSE department for this purpose.

The computers already in the lab might have enough processing power to process a limited number of image processing requests. However, this drastically reduces the scalability and reliability of our application. We might not be able to develop and test modules if we are not on the lab computers.

10 Assumptions

The following list contains critical assumptions related to the implementation and testing of the project.

- The team might purchase Amazon Cloud Storage and appropriate processing service for the application. There are alternatives like Microsoft Azure and Google Cloud and with further research, the team might decide either instead of AWS.
- All members have experience with Android development and some experience with computer vision.
- We will focus on clothing items to minimize the amount of module training, given we have limited time.
- We will use multi-threading to process images of interest while scanning other objects.
- The network bandwidth will be fast enough to seamlessly transmit images of about 4 Megabytes, for example.
- The $800.00 budget provided by the CSE department will suffice for all of our services including cloud hosting and any potential testing device we might decide to buy.

11 Constraints

The following list contains key constraints related to the implementation and testing of the project.

- Final prototype demonstration must be completed by May 1st, 2019
- Devices owned by customers may not have have the processing power to run our program
- Time to work on the project is limited due to the full-time course load of the members of the group
- Total development costs must not exceed $800
- Training the machine learning model will be very time-consuming

12 Risks

The following high-level risk census contains identified project risks with the highest exposure. Mitigation strategies will be discussed in future planning sessions.

Risk description	Probability	Loss (days)	Exposure (days)
Security flaws are found in the libraries in use	0.10	20	2
Testing devices do not function	0.01	5	0.05
Internet access is not available in Arlington, TX	0.01	1	0.01
APIs used to find pricing for items are down	0.01	1	0.01
Wrong budget estimation	0.2	5	0.1

Table 2: Overview of highest exposure project risks

13 Documentation & Reporting

13.1 Major Documentation Deliverables

13.1.1 Project Charter

The initial version will be delivered October 23rd, 2018. The final version will be completed sometime early next semester after all possible details have been decided on. This document will be maintained on a monthly basis unless something with the project changes drastically.

13.1.2 System Requirements Specification

The initial version will be delivered November 13th, 2018. The final version will be completed sometime in the Spring. The requirements will be updated whenever there is a requirement that needs to be added, deleted, or changed. It will be immediately updated whenever one of these conditions takes place.

13.1.3 Architectural Design Specification

The initial version will be delivered December 4th, 2018. The final version will be completed sometime in the Spring. This document will change only if core architectural design choices are changed. It will be immediately updated if this happens.

13.1.4 Detailed Design Specification

The initial version will be delivered February 26th, 2019. Any time the design specification changes this document will change, even if it is something minor since it is a detailed specification.

13.2 Recurring Sprint Items

13.2.1 Product Backlog

We will have grooming and iteration planning meetings to decide which items from the requirements we should put on the backlog for the iteration. They will be prioritized from highest to lowest starting at core functionality, defects, minor functionality, and aesthetics. The group can decide on major decisions for the items in the backlog. We will be using Wekan or another open source free software to keep track of the product backlog.

13.2.2 Sprint Planning

We will have our team meetings every Monday at 5:30 PM. We will plan each sprint the week before it starts during this meeting. There will be 8 sprints total between Senior Design 1 and 2. There are 4 sprints per semester each lasting 3 weeks in length.

13.2.3 Sprint Goal

The team will decide the sprint goal. We will involve the professor and may ask our friends about our current application for feedback.

13.2.4 Sprint Backlog

The backlog will be maintained via a scrum board. The group will decide what items will be on the backlog for that current sprint.

13.2.5 Task Breakdown

Each team member will claim tasks that they would like to work on. They will document the time spent on their tasks in their engineering notebook and then put this on their tasks as a note. The team members will create a to-do list for bigger tasks on the scrum board for each task they claim.

13.2.6 Sprint Burn Down Charts

Ivy will be responsible for generating the burn down charts for each sprint. Each team member will keep track of how much effort they spent on each item compared to the estimated amount of effort. The chart will show the amount of work left to accomplish compared to what the team is doing.

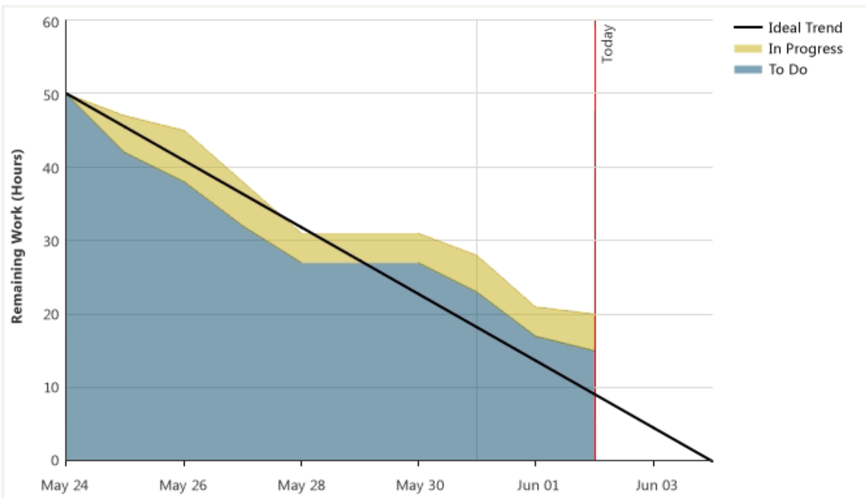

Figure 2: Example Sprint Burn Down Chart
[4]

13.2.7 Sprint Retrospective

The sprint retrospective will be handled at the end of each sprint during a meeting. It will be documented as a group and presented on the Friday labs when due.

13.2.8 Individual Status Reports

The individual status reports will include the work the individual has done during the current sprint, the plan to get done next sprint, and what unplanned work they had to take on. It will also list any problems and unresolved issues the individual has encountered. It will be reported every week.

13.2.9 Engineering Notebooks

The engineering notebooks will be updated every meeting, at minimum, by each team member. As many pages deemed necessary to each team member will be completed for each sprint. Each team member can sign and keep each other accountable by checking up every meeting.

13.3 Closeout Materials

13.3.1 System Prototype

The final system will be the application iSpy working on Android. The prototype will be demonstrated on an Android device on May 1st, 2019. The demonstration will use a few props that the application identifies.

13.3.2 Project Poster

The poster will be delivered sometime mid Spring semester. It will be 11x17 inches in size. The poster will include pictures of our app's functionality, logo, and a selling description of what our application does.

13.3.3 Web Page

The web page will contain a link to the Google Play store and information about the iSpy application. It will be accessible to the public through a public domain. The web page will be updated throughout the course of the semester, but will not be provided as complete until May 1st, 2019.

13.3.4 Demo Video

The demo video will be a basic walk-through for a typical user. The demo video will be uploaded to YouTube for better accessibility and will not be more than 2 minutes long. The video will guide the user how to download, install and run the application. Then the video will cover the most common use case of scanning an object of interest and finding information about the detected object.

13.3.5 Source Code

The source code will be maintained with the use of unit tests and code reviews throughout the lifespan of the project. Git will be the version control system used and only the executable/binary file will be given to the customer. The source code will not be open sourced to the general public.

13.3.6 Source Code Documentation

All methods will be annotated and classes will be annotated. Javadocs will be used to generate the documentation. The final documentation will be provided as a PDF.

13.3.7 Installation Scripts

The application will be available on the App Store as well as the Play Store for both Apple and Android devices. Both version of the application will include a guide for the user to learn off of if needed.

13.3.8 User Manual

We will provide users instructions on how to use the application within the application itself. There will be a page in the application that will teach the users the features of the application.

References

[1] Google Lens, 2018.

[2] THE ICONIC - Fashion Shopping, 2018.

[3] Britta OBoyle Chris Hall. Google Lens: What Is It and How Does It Work? Google Lens: What Is It and How Does It Work?, 2018.

[4] Doug Erikson. Sprint Burndown (Scrum) - TFS, 2018.

[5] Jen Karner. How to use Bixby Vision, 2017.

[6] Steven Winkelman. How to Use Bixby | Everything You Need to Know, 2018.