School of Computing
CA326 Year 3 Project Proposal Form

Project Title: Garden Lab

Student 1 Name: Anisa Hoxha	ID Number: 21413586
Student 2 Name: Zara Purewal	ID Number: 21404176

Staff Member Consulted: Dr. Alessandra Mileo

Project Description (1-2 pages):

Our proposed project is a garden planner web app. This app will be a tool that small to medium hobby gardeners can use to plan and monitor their gardens. It has three main features: a garden planning and layout tool, a plant well-being tool, and a plant identification tool.

The main feature of the web app is the virtual garden planning feature; Users will create a digital garden, where they will create a layout of their own virtual garden, a list of plants (or types of plants) they would like to include in a specific area (harder constraints) or anywhere in the garden (softer constraints). The user will also input other information such as location information as well as garden orientation, which can later be used to provide useful tips such as if a plant would be suitable depending on sunlight exposure, relevant alerts, such as disease outbreaks and adverse weather conditions. All the input information will be used to generate a suggested layout that maximises the user satisfaction and the health of the garden overall. Tools such as open source constraint satisfaction engines will be used for this purpose.

Once a virtual layout is proposed, the user can adjust or change it based on the app feedback, through the use of a drag and drop UI. Users can drag elements representing their desired plants and drop them where they would like to plant their plants relative to other plants. The app will provide useful information about each plant, such as optimal time to sow, how often to water and sowing distance.

The garden/pairs of plants will be rated either good, neutral, bad or unknown; based on a formula comparing a number of factors such as watering frequency, sunlight requirements, fertiliser frequency, and companion plants. (Information about companion plants will be found using web scraping if an API cannot be found for this purpose)

The virtual garden will be saved for a period of a month after last accessed. The user may retrieve the garden through their account. They can then edit/ tweak the garden as they wish. If the user decides to bring the virtual garden into reality, the status of the garden will be changed from “virtual” to “real”. A garden with the status of “real” will be saved for a period to two years after last accessed.

Once a garden (or section of a garden) is real, the user can then take photos of their garden to monitor, and photos will be kept in an online “diary”. From that time on, the user can use the second feature, the plant well-being tool. With this tool, users can take a picture of a plant, and using image recognition API, the app will provide a “diagnosis” for the plant. It could be deemed as healthy or it could be diagnosed; for example, if the leaves are yellow an “underwatered” diagnosis will be returned. This will allow the user to monitor their garden using the app.

The final feature, the plant identification tool, will allow the user to upload a picture of a plant, and using an image recognition API, receive the name and care instructions for the plant in question. If time allows, we will also implement Optical Character Recognition as an option to find the plant, where a user can scan the name of a plant to receive the same information.

Priority of Features

High

- Virtual garden planning which will include features such as compatible plants to place together, orientation of the garden, shade calculation, peak sunlight in morning vs evening (API work).
- Visualization of Planner, digital map of users garden.

Medium

- User uploads an image of the plant to identify the name of the plant & care instructions for that plant. (watering, ideal soil type)
- User uploads an image of the plant to identify what ailments/diseases it has and how to treat it.
- Have Alerts to suggest care information based on weather and location. (weather & google maps API)

Low/Time allowing

- Section for local gardening news, events, disease outbreaks, frost.
- Optical Character Recognition.
- New plant blog/ voting channel for users to suggest new features for Garden Lab.
- Security/ encryption of user data.

Division of Work - Outlines how the work is envisaged to be split equally among the team members.

Front-end

- Login page - Zara
- Drag and drop UI for garden planner - Anisa & Zara
- Photo diary - Anisa

Back-end

- Care instructions for each plant in garden planner using plant care info API - Anisa
- User information saving and Authentication – Zara

Functionality

- Virtual garden generation using constraint satisfaction engines - Anisa
- Plant/Garden compatibility score using a constraint satisfaction engine - Anisa
- Web scraping for info on compatible plants(Potential websites https://www.firsttunnels.co.uk/page/Companion-Planting-Guide, https://www.gardenersworld.com/plants/10-companion-plants-to-grow/, https://en.wikipedia.org/wiki/List_of_companion_plants#) - Zara
- Integration of plant image identification API - Zara

  - Plant well-being score based on an image of the plant- Zara
  - Optical Character Recognition for finding plant and care instructions - Anisa

Programming language(s) - List the proposed language(s) to be used

- Web App (Django, Node.js, React, Python)
- Joystick (UI framework with Node.js backend) https://github.com/cheatcode/joystick#what-is-joystick
- sortableJS - library that can be used with either React or Joystick, in order to make drag and drop UI https://sortablejs.github.io/Sortable/#multi-drag
- Rule Engine for Constraint Satisfaction- For example Minizinc https://js.minizinc.dev/docs/stable/ or Drools https://www.baeldung.com/drools

Programming tool(s) - List tools (compiler, database, web server, etc.) to be used

- Google Maps API, Weather API, Google Cloud API (for OCR)
- Database, to be decided, Firebase, MySQL, Google Cloud DB
- Data Sets (kaggle.com)
- Plant care info + disease info API https://perenual.com/docs/api
- Image Recognition API (Plant.id API)

Learning Challenges - List the main new things (technologies, languages, tools, etc) that you will have to learn

- Joystick and sortableJS
- Image Recognition API
- Web scraping
- Constraint Satisfaction Problem and engines

Hardware / software platform - State the hardware and software platform for development, eg. PC, Linux, etc.

- PC, Windows, on web browser

Special hardware / software requirements - Describe any special requirements

- N/A