TABLE OF CONTENTS

1. INTRODUCTION

1.1. About the Project

Within the scope of BM-314 Software Engineering semester project, it is planned to make an Object Detection application for the Visually Impaired to facilitate the lives of visually impaired individuals. The aim of the project is to facilitate the daily lives of visually impaired individuals by increasing their environmental awareness by being aware of the objects around them, their proximity and what they are.

The software to be created will consist of four parts, all of which will work in support of each other. First, an object detection AI model will determine whether there is an object around and, if so, where and what it is. Then a depth estimation model will estimate the distance to the found object. Finally, the visually impaired person will be informed about the found object, where it is located and how far away it is with an audio report.

As a result, the application will enable visually impaired individuals to recognize objects around them and facilitate their inclusion in daily life. This documentation covers the main features of the project, the process model and the assignment of team members.

1.2. Documents to be Delivered During Project Construction

Below are the documents and other elements targeted to be delivered within the scope of the project:

- Software Project Management Plan (SPMP)
- Software Requirements Specification (SRS)
- Software Design Description (SDD)
- Software Test Documentation (STD)
- Hotel Management System
- Presentation File

2. PROJECT ORGANIZATION

2.1. Software Process Model

In the project within the scope of BM-314 Software Engineering course, it was decided to use the Scrum Model, one of the software process models. This process model is used for frequent and

It has the advantages of fast meetings and easy division of labor for the requirements of the moment and fast action in uncertain situations. The most important factor that played a role in the choice here was the uncertainties in artificial intelligence models. The inability to find the data set required for a model, the inability to create a model that will work at the performance we want, or the need to take quick action if the plan will have to change if the working times are too long led us to choose this model. In addition to these, the structure of the Scrum model that requires constant communication with team members can be achieved thanks to the fact that we are 2 people and we are in constant communication. Considering these, it was decided to use the Scrum model.

2.2. Authorization and Responsibilities

Within the scope of the project, Furkan Egecan Nizam and Dilay Ece Maral aim to deliver the software on time, aiming to reach optimum solutions. There is no hierarchical structure in the project team; therefore, a project leader has not been identified. However, responsibilities were shared among the team members according to their areas of expertise. In this context, Furkan Egecan Nizam will be responsible for data science related work, while Dilay Ece Maral will be responsible for the rest of the software and development processes of the project.

2.3. Tools and Techniques to be used

The development of the artificial intelligence models of the related project will be carried out on Google Colab due to the high computational power requirement. In addition, the mobile backend and frontend design of the application will be done for the IOS platform via Swift. During the project development process, version control will be provided with Git and GitHub.

3. PROJECT MANAGEMENT PLAN

3.1. Tasks

The process of defining the tasks required defining the tasks with their uncertainties due to the high level of uncertainties in the project and the structure of the selected software development model that supports this. The modules that will be needed in the project are shown in Figure 3.1.1.

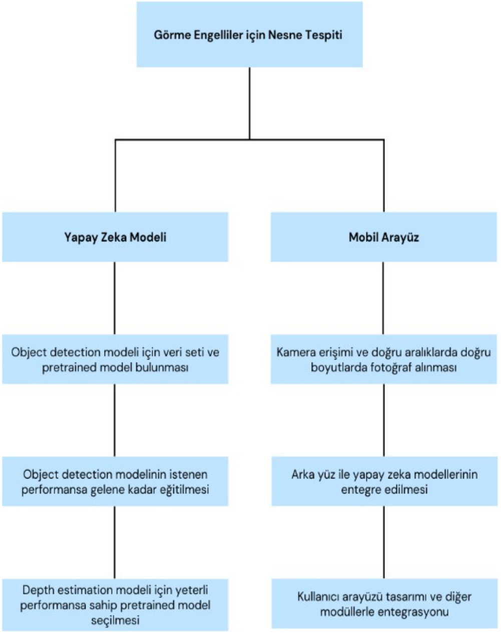
Görme Engelliler için Nesne Tespiti
Yapay Zeka Modeli
Mobil Arayüz
Object detection modeli için veri seti ve pretrained model bulunması
Kamera erişimi ve doğru aralıklarda doğru boyutlarda fotoğraf alınması
Object detection modelinin istenen performansa gelene kadar eğitilmesi
Arka yüz ile yapay zeka modellerinin entegre edilmesi
Depth estimation modeli için yeterli performansa sahip pretrained model seçilmesi
Kullanıcı arayüzü tasarımı ve diğer modüllerle entegrasyonu

Figure 3.1.1. Requirements Diagram

After determining the requirements, the next step is to determine how these requirements will be realized. Here, the development of the artificial intelligence models and the development of the mobile interface were to be carried out in parallel since they were carried out by two different people. The first reason for this was that the inputs and outputs of both the artificial intelligence model and the mobile interface were clear, even though they were not. First, the object detection model will take photographs; it will answer the questions of whether there is an object in these photographs, if so, where and what this object is. The necessary input for this, i.e. live video, will be provided from the phone's camera using the mobile interface. The output from the model will then be given to the depth estimation model, which will estimate how far away the detected object is, if any. When this information is available, it will be given to the mobile interface, and the mobile interface will transmit this information to the user in an audible way.

3.1.1. Object Detection Model

3.1.1.1. Description of the task

The object detection model is the backbone of the project. Model
It will continuously take pictures from the user's camera and analyze them,

will return whether there is an object in it, where it is and what it is.
This information is very cautious, as it covers the broad scope of what the project promises.
should be made. In addition, the requirements for continuous work and mobile working also necessitate a lightweight model.

3.1.1.2. Task steps

- Selecting the data set containing the objects required for the model
- Selecting a lightweight and high accuracy pretrained model
- Fine-tuning the selected model

3.1.1.3. Sources

3.1.1.3.1. Technical resources

- Kaggle
- Google Colab
- Hugging Face

3.1.1.4. Dependencies and restrictions

The object detection model is the cornerstone of the project. The rest of the project will be designed taking into account the good performance of this model. This is because the operation of the subsequent depth estimation model is dependent on this model, and the audio output to the user is dependent on the output of these two models. These factors cause all components of the project to depend on the object detection model. This model is subject to three constraints. The first one is the difficulty of finding a dataset that covers the generality of objects encountered in daily life. The second is that the size of the dataset is likely to be large, so even if a pretrained model is used, the fine-tuning process may take a long time, making it difficult to perform many trials. Lastly, the model needs to run on mobile devices, which requires high performance, which affects its accuracy due to its low computational power.

3.1.1.5. Identification of risks

As mentioned under dependencies and constraints, the object detection model is the foundation of the project. Therefore, taking appropriate precautions by predicting the errors that may be encountered while creating the model prevents the flow of the project from being disrupted and increases the quality of the software, since there is a plan B against the problems encountered. The risks and solutions foreseen to be encountered under this heading are as follows:

Risk1. Unavailability of the required data set.

Solution 1. Create a custom data set by collecting data containing the required objects from multiple data sets.

Risk 2. Taking too long to train the model.

Solution 2. Change the architecture of the model used, change the training logic of the model or train small samples from the dataset.

Risk3. High computational power requirement of the model.

Solution3. Downsize the model by converting it to a Tensorflow Lite model or converting it to 8 bits.

3.1.2. Depth Estimation Model

3.1.2.1. Description of the task

Depth estimation model, as the name suggests, depth detection
will do. In the case of the project, this will be used to determine how far away the objects detected by the object detection model are from the user.
will be used. When the object detection model detects an object, the model will run, and the depth of the image given to the object detection model as input will be used.
estimation model modified according to the input dimensions, and the output is the distance of the detected object.

3.1.2.2. Task steps

· Selecting a pretrained depth estimation model

· Performance tests on the selected model

· Fine-tune the model if necessary

3.1.2.3. Sources

3.1.2.3.1. Technical resources

· Google Colab

· Hugging Face

· Kaggle

3.1.2.4. Dependencies and restrictions

How we react to the objects we encounter on our way from one place to another is not just a matter of

It doesn't matter what the object is and which side it is on, one of the most important factors here is the distance of the object. For a nearby object, we need a sudden and sharp reaction. In this project, the distance information to be provided to the visually impaired individual is the responsibility of this model. The most important dependency and constraint of the model is the output of the object detection model that will run before it. Because the model will inform the user about the distance of the place that the object detection model defines as an object. The dependency of the project on this model is that a very important point in the output to the user is dependent on this model. Another limitation of the model is that it needs to work with high performance on mobile devices with low computing power.

3.1.2.5. Identification of risks

The most prominent risk of the model is that it is dependent on the object detection model at the very beginning. In addition, since this model is also an artificial intelligence model, errors that can be seen in the object detection model can also be seen here.

Risk1.The output of the object detection model is incorrect.

Solution1. Ensure the performance of the object detection model before proceeding to this step

Risk2. The model takes too long to run.

Solution 2. Implement performance enhancements such as converting the model to Tensorflow Lite and 8-bit, or switch to a mathematical computation-based approach instead of artificial intelligence.

Risk3. Failure to obtain the desired results from the models.

Solution3. Find data sets close to the possible inputs in the project and fine-tune the models.

3.1.3. Mobile Interface

3.1.3.1. Description of the task

First, images should be taken from the camera of the mobile device at certain intervals, these should be converted into the format accepted by the object detection model, which is the first model, and then given to the object detection model. After that, the results of the object detection model should be checked, and if an object is found, the image should be converted into the format accepted by the depth estimation model, and then given to the depth estimation model. The outputs of these two models should be formatted to be given to the user, and then this information should be transmitted to the user by voice.

3.1.3.2. Task steps

· Taking images from the mobile device's camera at certain intervals

· Formatting the received images according to the object detection model

· Interpretation of the output of the object detection model

· Formatting the image according to the depth estimation model

· Formatting the printouts to be taken and given to the user

· Voice delivery of information to the user

3.1.3.3. Sources

3.1.3.3.1. Technical resources

· Swift
· AVFoundation (for camera access and image processing)
· Core ML (for integration of AI models)
· Text-to-Speech (for voice feedback)
· Xcode (Development environment)

3.1.3.3.2. Written sources

· IEEE Standards

· Apple Developer Documentation

3.1.3.4. Dependencies and restrictions

Camera access and acquisition of images, which are the source of the models' inputs, are the

This interface is responsible for formatting the images in accordance with the input dimensions of the models, connecting the models to each other, the control that needs to be done during this process, formatting the outputs of the models and notifying the user audibly. As can be seen, the entire project is dependent on this interface, so it is the skeleton of the project. Besides, it has no direct impact on the performance of the models, which limits the performance of the interface to the performance of the models.

3.1.3.5. Identification of risks

The mobile interface that connects everything in the system ensures that all operations such as camera access, image processing, formatting of model outputs and voice feedback are performed accurately.
in an orderly and error-free manner. An error in the interface can lead to problems such as failure to process images, misinterpretation of model outputs or delayed audio feedback, making the system inaccurate or unnecessary.
may cause it to work.

Risk1. Failure to obtain images due to camera access problems.
Solution1. Ensure camera permissions are configured correctly and device compatibility. Test AVFoundation configuration and troubleshoot possible access issues.

Risk2. Failure to convert the received images into a format suitable for object detection and depth estimation models.
Pre-test format conversions and establish error management mechanisms for format incompatibilities.

Risk3. Model output formats cannot be formatted in a user-friendly way. Solution3. A standard structure for output formats should be determined and the format conversion process should be automated.

Risk4. Delay or incorrect vocalization during audio feedback with Text-to-Speech.
Solution4. Test Text-to-Speech configuration and use buffering mechanism for delays.

Risk6. Overheating of the device or rapid battery drain during operation.

Solution6. Optimize the processing load and remove the code that creates unnecessary processing load in the background.

Risk7. Incompatibility after Swift, AVFoundation or Core ML version updates.
Solution7. Version updates should be checked in a test environment before release updates and updates should not be made without ensuring compatibility.

3.2. Assignment

Furkan Egecan Nizam is responsible for the data science part of the project and Dilay Ece Maral is responsible for mobile interface development.

3.3. Time Schedule

Although it would not be correct to give an exact timeline due to the nature of the Scrum software development model, an estimated timeline is given in Table 3.3.1.

History	**Mission**	**Responsible Employee**
21.03.2025 / 28.03.2025	Deciding on the required data set and pretrained model for the object detection model	Furkan Egecan Nizam
18.03.2025 / 21.03.2025	Resolving camera access	Dilay Ece Maral
28.03.2025 / 04.04.2025	Fine-tuning the object detection model with the required data set	Furkan Egecan Nizam
28.03.2025 / 04.04.2025	Solving the voice output feature	Dilay Ece Maral
04.04.2025 / 11.04.2025	Deciding on the pretrained model to be used for depth estimation	Furkan Egecan Nizam
04.04.2025 / 11.04.2025	Tests to take photos from the camera and run the object detection model	Dilay Ece Maral
16.04.2025	Software Requirements Specification Delivery	Furkan Egecan Nizam / Dilay Ece Maral

11.04.2025 / 18.04.2025	Performance improvements for the object detection model based on feedback	Furkan Egecan Nizam
11.04.2025 / 18.04.2025	Bridging the gap between object detection model and depth estimation model	Dilay Ece Maral
18.04.2025 / 25.04.2025	Performance improvements for the Depth estimation model based on feedback	Furkan Egecan Nizam
18.04.2025 / 25.04.2025	Formatting the received printouts and transmitting them to the user audibly	Dilay Ece Maral
25.04.2025 / 02.05.2025	Full system tests	Furkan Egecan Nizam / Dilay Ece Maral
30.04.2025	Software Design Description Delivery	Furkan Egecan Nizam / Dilay Ece Maral
02.05.2025 / 09.05.2025	Performance improvements based on tests	Furkan Egecan Nizam / Dilay Ece Maral
09.05.2025 / 14.05.2025	Latest system tests	Furkan Egecan Nizam / Dilay Ece Maral
14.05.2024	Software Test Documentation Delivery	Furkan Egecan Nizam / Dilay Ece Maral
~	Project Presentation	Furkan Egecan Nizam / Dilay Ece Maral

Table 3.3.1. Timeline

4. ADDITIONAL MATERIALS

There is no additional material required under this heading.