**ATILIM UNIVERSITY**

Software Engineering Department

SE 494

Senior Project Proposal for Cooperative Education

# RTSG

Proposed by

SAMET ÇOLAK

Moderated by

MURAT AYKAN

AYESAŞ

Coordinated by

ALİ YAZICI

2024-2025 Fall

## Foreword

IMPORTANT, READ FIRST

***Do not remove or modify the sections with these notations in any way.***

*All guidelines in this template appear in this 'boxed' format. These instructions, as well as the preceding title page and this foreword, should never be removed from the submitted files.*
*All texts between "<" and ">" symbols (incl. on the front page and in the headers) should be replaced or removed.*
*Please comply with the formatting rules by adhering to the following guidelines:*

- *Do not remove the instructions.*
- *The proposal size should be less than 3 MB.*

## Change log

*The change log is a table which provides an overview of changes made in the Project Proposal Document. A description of the changes must be provided in a concise manner.*

| Version | Submission date | Description of changes | Affected parts |
|---|---|---|---|
| V1 | 25/09/2024 | Added project description and key tools. | Section 2<br>Section 3 |
| V2 | 26/09/2024 | Added project plan and deliverables list. | Section 4<br>Section 5 |
| V3 | 27/09/2024 | Added project management. | Section 6 |
| V4 | 30/09/2024 | Added acronyms and project key data, reviewed the report from start to finish. | All Section |

## Table of Contents

## Acronyms

| | |
|---|---|
| RTSG | Radar and Track Scenario Generator |
| GIS | Geographic Information Systems |
| IDE | Integrated Development Environment |
| GUIs | Graphical User Interfaces |
| CI | Continuous Integration |
| CD | Continuous Deployment |
| JDK | Java Development Kit |
| WP | Work Package |
| TDD | Test-Driven Development |
| UCD | User-Centered Design |
| IP | Intellectual Property |

# 1 Project Key Data

***Project Short Title*** *is a short name of the project, if it exists. For example "NoFaDe" or "NFDP", for a project having Full Title of "Novel Face Detection Platform Development".*

***Subject Classification*** *is the general Computer Engineering/Software Engineering/Information Systems Development area the project can be classified in, e.g. image recognition, software testing, Big Data, Artificial Intelligence, Web-based Application, Mobile Application, Game Application, IoT-based Smart Application, Cloud-based Application, Software Security, Cybersecurity, etc.*

| | |
|---|---|
| Project Short Title | RTSG |
| Project Full Title | Radar and Track Scenario Generator |
| Subject Classification | Simulation and Modeling |
| Student name(s)<br>(Team members of project) | Samet Çolak |
| Full Name of Corporation | AYDIN YAZILIM ve ELEKTRONİK SANAYİ A.Ş. |
| Mentor Name(s)<br>(Corporation member) | MURAT AYKAN |
| Coordinator Name(s)<br>(University member) | ALİ YAZICI |

# 2 Project Description

*In this section, you are to provide information in both English and Turkish.*

*Within a maximum of one page, provide a brief description of the project that you are proposing. Your descriptions should include the problem you are addressing, the aim and the scope of your project.*

*En fazla bir sayfa içinde, projenizle ele aldığınız sorunun kısa bir tanımını yapınız. Açıklamalarınız projenizin amacını ve kapsamını içermelidir.*

## 2.1 Project Short Description

Radar and Track Scenario Generator is an innovative software application that simulates data obtained from radar systems and generates track information in a text file. This project allows users to add radars and tracks, enabling real-time or accelerated visualizations. RTSG offers a user-friendly interface for radar setup, allowing users to configure radars with features such as latitude, longitude, radar name, scanning radius, starting angle, and error factor. The track creation process is performed by determining the start and end points, with the direction calculated automatically. Additionally, while the project initially uses a PNG format background map, future plans include the integration of a 2D map supported by Geographic Information Systems (GIS). Users can effectively observe the scanning areas of radars and track information, and save this data to a text file. RTSG aims to enhance data accuracy in the defense industry by mimicking radar behaviors.

## 2.2 Projenin Kısa Tanımı

Radar ve Track Scenario Generator , kendi radar ve track'lerini oluşturarak bu tracklarin bilgilerini bir metin dosyasında oluşturan yenilikçi bir yazılım uygulamasıdır. Bu proje, kullanıcıların radar ve track ekleyerek gerçek zamanlı veya hızlandırılmış görselleştirmeler yapmalarına olanak tanır. Radar kurulumu için kullanıcı dostu bir arayüz sunan RTSG, enlem, boylam, radar adı, tarama yarıçapı, başlangıç açısı ve hata faktörü gibi özelliklerle radarları yapılandırabilmeyi sağlar. Track oluşturma işlemi, başlangıç ve bitiş noktalarının belirlenmesiyle gerçekleştirilir ve yön otomatik olarak hesaplanır. Ayrıca, proje başlangıçta bir PNG formatında arka plan haritası kullanırken, ilerleyen süreçte Coğrafi Bilgi Sistemi (CBS) destekli 2D bir harita entegrasyonu planlanmaktadır. Kullanıcılar, radarların tarama alanlarını ve track bilgilerini etkili bir şekilde gözlemleyerek, bu verileri bir metin dosyasına kaydedebilirler. RTSG, radar davranışlarını taklit ederek, savunma sanayi alanında veri doğruluğunu artırmayı amaçlamaktadır.

## 3 Key Tools, Techniques and Technologies

*In a maximum of two pages, outline the key tools, techniques, and technologies along with their intended use. For example;*
*- “At least three alternative Convolutional Neural Network models will be used for face identification.”*
*- Pretrained models will be used for Transfer Learning purposes.*
*- “Python language would be used to develop Jupyter Notebook files containing the solution that utilizes PyTorch Framework.”*
*- Eclipse IDE will be used in coding, debugging and execution.*
*- The Gitlab environment would be used for version tracking, Continuous Integration (CI) and Continuous Deployment (CD) purposes.*
*- etc.*

1. **Programming Language: Java**

Java will be the primary programming language used to develop the SRTSG application. Its platform independence and robustness make it an ideal choice for building software.

2. **Integrated Development Environment (IDE): Eclipse**

Eclipse IDE will be utilized for coding, debugging, and executing the Java application. It provides a comprehensive development environment with features like code completion, syntax highlighting, and integrated debugging tools, enhancing productivity during the development process.

3. **JavaFX for User Interface**

JavaFX will be employed to create the user interface of the application, providing a modern and responsive design. This framework allows for the creation of rich graphical user interfaces (GUIs) that improve user experience when interacting with the radar and track simulation features.

4. **File I/O for Data Handling**

Java's built-in File I/O capabilities will be used to read from and write to text files. This will enable the application to log radar and track data generated during the simulation, allowing users to store and retrieve their configurations and results effectively.

5. **Geographical Information Systems (GIS) Integration**

Although the initial version will use a PNG format map, plans are in place to integrate GIS technology for future enhancements. This integration will allow for more accurate geographical representations and improved radar tracking capabilities.

### 6. **Version Control: GitLab**

The GitLab platform will be employed for version tracking, allowing for effective collaboration, code management, and version control throughout the development process. This will facilitate Continuous Integration (CI) and Continuous Deployment (CD) practices, ensuring that the application is always in a deployable state.

### 7. **Simulation Techniques**

The project will utilize simulation techniques to model the behavior of radar systems and track objects in real-time. This will include algorithms for calculating the trajectory of tracks based on user-defined parameters such as velocity, course, and radar scanning radius.

### 8. **Real-Time Visualization**

The application will implement real-time visualization techniques to display the radar's coverage and the tracked objects on the user interface, providing users with immediate feedback on the simulation results.

# 4 Project Plan

*In this section you are to provide information about your project plan, including the methodology (e.g. Waterfall, Scrum, Kanban, DSDM, etc.) that you would be following, the Time Schedule (Gantt Chart) of your main Work Packages (or Tasks), and the Deliverable List with their due dates. Be careful that you are expected to obey the deadlines you indicate, and produce the outcomes that you list under the deliverable list.*

## 4.1 Methodology

*Explain the system/software development methodology to be used such as waterfall, rapid application development, scrum, kanban, etc. Your methodology may also include a suitable Research Methodology, if appropriate for your project. Also, give information about the methodology to be used.*

The RTSG project will utilize the Incremental Methodology for its development. This approach allows for the project's gradual development through multiple iterations, enabling regular evaluations and improvements at each stage.

Key Features of Incremental Methodology:

**Phased Development**: The project will be divided into different phases, focusing on specific aspects of the application. This allows the development team to concentrate on particular functionalities and improve them incrementally.

**User Feedback**: After each increment, user feedback will be collected to verify that the evolving features meet the targeted requirements and enhance the user experience.

**Risk Management**: This methodology enables early identification of potential risks and issues, allowing for timely interventions and adjustments in the project plan.

**Continuous Integration**: Each increment will be integrated into the existing system, ensuring that all features work together harmoniously before the final release.

**Testing Process**: Regular testing will be conducted at each stage of the development process to ensure that new features work correctly and do not adversely affect existing functionality.

Overall, the Incremental Methodology will provide a structured yet flexible framework for the successful development of the RTSG project, ensuring that it meets its objectives and delivers value to users.

## 4.2 Time Schedule (Gantt Chart)

*Gantt Chart timing may be based on either months or weeks and you should either use Work Package or Task for the items, whichever is appropriate for your project plan. The content of the following gantt chart is given as an example and you should replace them with yours.*

| WP / Task No | WP / Task Name | Start Date | End Date | Weeks | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 |
| WP0 / T0 | Project Management | 20/08/2024 | 10/01/2025 | ▨ | ▨ | ▨ | ▨ | ▨ | ▨ | ▨ | ▨ | ▨ | ▨ | ▨ | ▨ | ▨ | ▨ | ▨ | ▨ | ▨ | ▨ | ▨ | ▨ | ▨ |
| WP1 / T1 | Requirement Analysis | 20/08/2024 | 26/08/2024 | ▨ | | | | | | | | | | | | | | | | | | | | |
| WP1 / T2 | Design | 26/08/2024 | 09/09/2024 | | ▨ | ▨ | | | | | | | | | | | | | | | | | | |
| WP1 / T3 | Development | 09/09/2024 | 18/11/2024 | | | ▨ | ▨ | ▨ | ▨ | ▨ | ▨ | ▨ | ▨ | ▨ | ▨ | ▨ | | | | | | | | |
| WP1 / T4 | Testing | 11/11/2024 | 25/11/2024 | | | | | | | | | | | | ▨ | ▨ | ▨ | | | | | | | |
| WP1 / T5 | Feedback | 25/11/2024 | 02/12/2024 | | | | | | | | | | | | | | ▨ | ▨ | | | | | | |
| WP2 / T1 | Requirement Update | 02/12/2024 | 09/12/2024 | | | | | | | | | | | | | | | ▨ | ▨ | | | | | |
| WP2 / T2 | Design | 09/12/2024 | 16/12/2024 | | | | | | | | | | | | | | | | ▨ | ▨ | | | | |
| WP2 / T3 | Development | 16/12/2024 | 06/01/2025 | | | | | | | | | | | | | | | | | ▨ | ▨ | ▨ | ▨ | |
| WP2 / T4 | Testing | 06/01/2025 | 09/01/2025 | | | | | | | | | | | | | | | | | | | | ▨ | ▨ |
| WP2 / T5 | Feedback | 09/01/2025 | 10/01/2025 | | | | | | | | | | | | | | | | | | | | | ▨ |

## 5 Deliverables List

*Think about the artifacts that you will produce during your proposed project and the methodology that you will follow up and list the common deliverables with due dates. The type of deliverable can be either Software, Module, Document, or Others.*

*List the intermediate and final deliverables of the project. The number and type of your reports should be determined considering this list.*

*Following table contains example items, and should be replaced with your own list.*

| ID | Type | Description | Due Date |
|---|---|---|---|
| **D1** | Document | Project Plan: Comprehensive plan outlining project scope, timeline, and resources. | 20/08/2024 |
| **D2** | Document | Requirements Specification: Detailed specifications of system requirements. | 27/08/2024 |
| **D3** | Document | Design Document: Architectural design and design patterns used. | 10/09/2024 |
| **D4** | Software | Prototype of Radar Detection System: Initial version of the radar detection functionality. | 18/11/2024 |
| **D5** | Document | Test Plan: Plan detailing the testing strategy and scope. | 02/12/2024 |
| **D6** | Software | First Build of Simulator: First complete build of the simulator for review. | 02/12/2024 |
| **D7** | Document | Feedback Report: Report summarizing feedback received from the first build. | 09/12/2024 |
| **D8** | Document | Updated Requirements Document: Revised requirements based on feedback. | 16/12/2024 |
| **D9** | Software | Final Build of Simulator: Final version of the simulator, incorporating all updates and fixes. | 06/01/2025 |
| **D10** | Document | User Manual: Comprehensive guide for users on how to operate the simulator. | 07/01/2025 |
| **D11** | Document | Final Project Report: Detailed report summarizing project outcomes, lessons learned, and future recommendations. | 10/01/2025 |

# 6 Project Management

## 6.1 Risk Management

*The risks that may adversely affect the success of the project and the measures to be taken to ensure the successful execution of the project when these risks are encountered (Plan B) should be outlined in the Risk Management Table. More rows can be added.*

| | **Risk Description** | **Response Action (B Plan)** |
|---|---|---|
| 1 | Uncertainty of requirements | Conduct workshops to clarify the requirements. |
| 2 | Technical issues (software bugs) | Increase the frequency of software testing and set up a backup development environment. |
| 3 | Risk of not completing on time | Review the project timeline and prioritize tasks. |
| 4 | Absenteeism of team members | Identify backup team members and create a flexible work plan. |
| 5 | Delays in customer feedback | Enhance communication by scheduling regular feedback meetings. |
| 6 | Budget overruns | Establish budget monitoring mechanisms to regularly check expenses. |

## 6.2 Budget Estimation

*Within this section you are to calculate the estimated cost of the project, constituting effort, and other costs (such as license, tools, libraries, etc.).*

### 6.2.1 Effort Estimation (person per month)

*You should estimate manpower effort requirements for each Task/WP given in the Time Schedule, and sum them up to calculate total requirement in man-hour or man-month.*

In estimating the budget for the project, I have calculated the total man-hours required for each task based on the scheduled work weeks and my daily working hours. The breakdown of the man-hours for each task is as follows:

WP1 / T1 Requirement Analysis: 45 hours (5 days * 9 hours/day)

WP1 / T2 Design: 81 hours (9 days * 9 hours/day)

WP1 / T3 Development:  hours (44 days * 9 hours/day)

WP1 / T4 Testing: 80 hours (9 days * 9 hours/day)

WP1 / T5 Feedback: 40 hours (5 days * 9 hours/day)

WP2 / T1 Requirement Update: 40 hours (5 days * 9 hours/day)

WP2 / T2 Design: 40 hours (5 days * 9 hours/day)

WP2 / T3 Development:  80 hours (10 days * 9 hours/day)

WP2 / T4 Testing: 40 hours (5 days * 9 hours/day)

1. WP2 / T5 Feedback: 8 hours (1 day * 8 hours/day)

Total Man-Hours: 45+81+400+80+40+40+40+80+40+8=854 hours

Assuming a standard work month consists of approximately 160 hours (40 hours/week * 4 weeks).

Total Estimated Man-Months: 854 hours / 160 (hours/month) ≈ 5.34 months

The estimated total cost of the project, including all aspects such as personnel expenses, overhead, and general expenditures, is approximately 300,000 TL. This total encompasses various costs such as intern salaries, overhead expenses, general expenses, and profit margins. Due to company policy, I am unable to provide a detailed breakdown of these costs.

Overall, the budget reflects the anticipated human resource costs and indicates that all necessary financial provisions have been made to support the successful completion of the project.

### 6.2.2 Other Project Requirements

*List any other computer, HW, SW, license, cloud service, etc. needed during project development. Specify whether the requirement is available or not. If it is not available, write its price and calculate the total cost of materials to be purchased. More rows can be added if needed.*

| **Id** | **Name of requirements** | **Its usage in the project (why you need it)** | **Available /<br>To be purchased** | **Its cost (if it needs purchasing)** |
|---|---|---|---|---|
| 1 | Computer Hardware | Necessary for developing and testing the simulator. | Available | - |
| 2 | Java Development Kit (JDK) | Essential for Java development. | Available | - |
| 3 | Integrated Development Environment (IDE) (e.g., Eclipse) | Required for coding and debugging. | Available | - |
| 4 | JavaFX Library | Necessary for creating the user interface of the project. | Available | - |
| 5 | Software Licenses | For any additional tools that may be needed (e.g., database, APIs). | Available | - |
| 6 | Testing Tools | Necessary for unit and integration testing. | Available | - |
| Total | | | | - |

## 6.3 Standards

*Specify the standards to be followed during the project.*

During the development of the Radar and Track Scenario Generation Simulator project, the following standards will be adhered to in order to ensure quality, consistency, and compliance throughout the project lifecycle:

1. **Programming Standards**
   - **Java Code Conventions:** Follow official Java coding conventions to ensure readability and maintainability of the code, including naming conventions, indentation, and commenting practices.
   - **Documentation Standards:** Use JavaDoc for documenting public classes, methods, and fields to ensure comprehensive documentation for future reference and for other developers.
2. **Development Process Standards**
   - **Incremental Methodology:** The project will utilize an incremental approach, focusing on building and refining the simulator in stages. Each increment will deliver a usable version of the software, allowing for ongoing feedback and adjustments based on evaluations and testing results.
3. **Collaboration and Support**
   - **Task Management:** The project tasks and progress will be tracked using Jira, enabling efficient management and documentation of project workflows, task assignments, and deadlines.
   - **Team Collaboration:** As the project is developed individually, I will rely on Jira to track issues, manage tasks, and ensure organized communication regarding any challenges. Additionally, collaboration with other employees will be sought when needed for problem-solving.

4. **Testing Standards**

    - **Test-Driven Development (TDD):** Implement TDD practices to create automated tests before coding features, ensuring that all functionalities are thoroughly tested and validated.

    - **Unit Testing Framework:** Use JUnit as the primary unit testing framework to validate the functionality of individual components and ensure the reliability of the codebase.

5. **Quality Assurance Standards**

    - **Version Control Process:** Utilize Git for version control, hosted on Bitbucket. All changes will be tracked through commits with detailed commit messages, ensuring transparency and a clear history of modifications.

    - **Continuous Integration (CI):** Bitbucket Pipelines can be integrated for CI to automate building, testing, and deployment processes. This ensures that every code change is seamlessly integrated and tested, minimizing the risk of bugs.

6. **User Interface Standards**

    - **User-Centered Design (UCD):** Follow UCD principles to create a user-friendly interface, focusing on usability and accessibility for diverse users.

    - **JavaFX Design Guidelines:** Adhere to JavaFX design guidelines to ensure a consistent and responsive user interface.

7. **Security Standards**

    - **Secure Coding Practices:** Follow secure coding guidelines to mitigate risks related to security vulnerabilities (e.g., input validation, error handling).

8. **Documentation Standards**

    - **Version Control Documentation:** Document changes using Git with detailed commit messages. Version control will be managed through Bitbucket, and each change will be clearly recorded with proper descriptions in both Git commits and the Bitbucket repository, ensuring proper tracking and accountability for all updates.

## 6.4 Ethical and Legal Requirements / Issues

*1. Describe any potential ethical issue that may arise during execution of the project. Describe your strategies to be followed to reduce ethical risk.*

*2. Indicate if there are any legal risks that may be encountered during the project implementation or during the commercialization of the project output.*

### 1. Ethical Issues and Strategies to Reduce Ethical Risk

The development of the Radar and Track Scenario Generation Simulator project involves potential ethical issues, particularly concerning privacy and data usage. The project may handle sensitive information related to radar operations and simulated tracking of objects, which could lead to ethical concerns about how this data is collected, stored, and utilized.

**Strategies to Reduce Ethical Risk:**

- **Data Anonymization:** Ensure that any data collected during simulations is anonymized to protect the identity and privacy of individuals or organizations involved. This means removing any identifiable information before analysis or reporting.
- **Transparent Communication:** Clearly communicate the purpose of data collection and usage to all stakeholders. This will help build trust and ensure that everyone involved understands the project's objectives and methodologies.
- **Adhering to Ethical Guidelines:** Follow established ethical guidelines for research and development in the defense and simulation sectors. This includes ensuring compliance with ethical standards set by relevant professional organizations.
- **Regular Ethical Reviews:** Conduct regular reviews of the project's practices to identify and address any ethical concerns that may arise during development. Engaging an ethics board or committee can help provide insights and guidance on best practices.

### 2. Legal Risks

Legal risks may arise during the implementation of the project and its eventual commercialization, including:

- **Intellectual Property (IP) Issues:** The project may involve the use of proprietary algorithms, software libraries, or technologies. It is essential to ensure that proper licenses are obtained for any third-party resources utilized in the development process to avoid potential infringement on IP rights.
- **Compliance with Regulations:** The project must comply with local, national, and international regulations governing the use of radar technology and data collection, particularly in defense and surveillance contexts. Non-compliance could result in legal penalties or restrictions on project deployment.

**Mitigation Strategies:**

- **Consultation with Legal Experts:** Engage with legal counsel experienced in intellectual property and technology law to navigate potential risks and ensure compliance with all relevant laws and regulations.
- **Regular Legal Audits:** Conduct regular audits to ensure that the project adheres to all applicable legal requirements and best practices in data protection and technology use.
- **Documentation and Compliance Checks:** Maintain thorough documentation of all software licenses, data usage policies, and compliance measures taken throughout the project lifecycle.

By proactively addressing these ethical and legal considerations, the RTSG project aims to minimize risks and ensure responsible development and deployment.