1. AI training:

- Collect a dataset of malware and benign samples. This dataset should be as large and diverse as possible to ensure that the trained model is generalizable.
- Preprocess the dataset.This may involve cleaning the data, removing outliers, and converting the data to a format that is compatible with the chosen machine learning framework.
- Select a machine learning algorithm. There are many different machine learning algorithms that can be used for malware detection and classification. Some popular choices include support vector machines (SVMs), random forests, and deep neural networks (DNNs).
- Train the machine learning model. This involves feeding the preprocessed dataset to the chosen machine learning algorithm and allowing it to learn the patterns that distinguish malware from benign samples.
- Evaluate the trained model. This involves testing the trained model on a held-out dataset to assess its performance.

2. Coding and development:

- Design the system architecture. This involves determining how the different components of the system will interact with each other.
- Implement the system using the selected technologies. This may involve writing code to load the trained model, preprocess new data, and generate predictions.
- Unit test the individual components of the system This involves testing individual units of code, such as functions and classes.
- Integration test the system as a whole.This involves testing how different components of the system work together.

Testing

1. Write a test plan that describes the different types of tests that will be performed.

- Functional testing:This involves testing the system's functionality to ensure that it meets the requirements.
- Performance testing: This involves testing the system's performance under different loads.
- Security testing:This involves testing the system for security vulnerabilities.

2. Write test cases for each test.

3. Execute the test cases and fix any bugs that are found.

Deployment

1. Write a deployment plan that describes how the system will be deployed in the production environment.

- Identify the target environment. This includes determining the hardware and software requirements for the system.
- Develop a deployment script. This script should automate the process of deploying the system to the target environment.
- Test the deployment script in a staging environment. This will help to ensure that the deployment process is smooth and error-free.

2. Create training materials for the users.

3. Provide support for the system using a ticketing system such as Jira or Zendesk.