

# Software Project Report Outline

## 1. Team Information

Team Name: Team 17

Team Members: 2

Member 1: Ege Oztas

Email 1: ge85jon@mytum.de

Member 2: Loris Nathan Verga

Email 2: go72jak@mytum.de

Module: Distributed Hash Table sub-module

Implementation: Kademlia

## 2. Programming Language and Operating System

We have chosen Python as our projects Programming language. As it provides multitude of libraries and adaptability to any need of the project. We believe with python's volumes of libraries our workload would be minimized from the start. There is also the experience and familiarity with the language of one of our members being experienced in both python and low level python web development.

For the operating system Ege will be using windows machine to develop the project. Because of the convenience it provides with daily academic life path of least resistance for him was to develop our project among the digital storage paces of his academic notes and other projects for other courses.

Loris will be using Linux Ubuntu 23.10 to develop the project because he had been working with it on a daily basis for several years, and because of the ease with which the development environment can be configured.

To ensure compatibility, we are going to use the python virtual environment. In addition, this will facilitate deployment for the test phases. We are considering using docker containers configured to run Python applications, to facilitate testing when multiple instances of the project need to be running at the same time.

### 3. Build System

Python is an interpreted language, and does not require compiling. To make it easier to test multiple instances of peers, we are going to write scripts to create, launch and monitor docker containers containing the multiple processes.

### 4. Quality Measures

For software testing Python offers variety of libraries. We are planning to use Pytest for our test cases with DHT. Pytest offers us easy fixture interfaces and verbose test results. We also had previous experience with using pytest for software testing purposes.

#### Test Cases

We plan to use a cooperative testing strategy. Each member will be responsible for testing the other member's code. Each function must be fully tested, including edge cases.

#### Quality Control Tools

To ensure a consistent and clean code style, we have chosen to use the flake8 lint tool. Flake8 is open source and perform static source code analysis and check for symantec discrepancies.

### 5. Libraries

- Socket: The python library to work with network socket logic. Allows us to create connections between different network endpoints. Is essential for our project.
- Threading: Allows python to easily work with multithreading logic. Will use it to work on the different peers of the network if need be.
- Hashlib: The main hashing library for python. We will use it to create hashes and hash tables.
- pytest: A testing enviroment for Python. Will use it to seperately run testcases for our DHT project.
- time: A library to work with time in detail in Python

It should be noted that this short list of libraries are just the initial planned ones and if it is needed to use another library from pythons libraries we would expand this list in the future reports.

### 6. License

Based on Kerckhoffs's principle, security should not be based on the obscurity of source code. For this reason, we chose an open source license that would allow external operators to audit our code. We also support the development of open-source software, and for this reason we choose the MIT license, which is fairly unrestrictive for possible improvements by external contributors.

## 7. Previous Experience

Experience:

- Ege Öztaş: Worked with python last 3 academic years. Developed backend of a video conference and education startup using Pythons Django framework. Used python for both Network Security and Security Engineering Classes' homework assignments given by Prof. Carle. For the last 1 year worked with a Cybersecurity firm as python test developer as a part time student worker.
- Loris Nathan Verga: Worked with python for machine learning, deep learning and network security course and some for scripting purposes during industry internships. Created a google maps-style bike route planner in Java as part of an object-oriented programming project. Worked 6 months full time as a systems engineer intern in a logistics company.

## 8. Workload Distribution

Distribution Plan:

<b>Task</b>	<b>Deadline</b>	<b>Person</b>
Getting to know and understanding Kademia implementation.	3.05.24	Loris and Ege
Read Kademia Paper	3.05.24	Loris and Ege
Implement a network interface on the python script to be built on by the other Functions	11.05.24	Ege
Definition of the distance metric for Kademia	11.05.24	Loris
Creation of the data structure that stores the k-buckets + interface.	11.05.24	Loris
Creation the k-bucket filling algorithm.	18.05.24	Loris
Implementing the key allocation/re-publishing functionality of Kademia.	25.05.24	Ege
Defining requirements and abstractions of Project specified functions (PUT, GET, SUCCESS, FAILURE) and planning the Kademia framework's functions under these abstractions.	18.05.24	Ege
Implementing Planned Functions. Will be detailed further in the following reports.	02.07.24	Loris and Ege
Testing the functionality with the dummy implementations provided for the final evaluation.	02.07.24	Loris and Ege
Performance optimization: caching of K, V pairs	02.07.24	Loris and Ege