SOEN 6431
SOFTWARE COMPREHENSION AND MAINTAINANCE

Concordia University

Department of Computer Science and Software Engineering

Deja Vu - Deliverable 1

Authors (Team G):

Snehee patel Yash Nareshbhai Radadiya

Sadath Roshan Parth Shah

June 12, 2023

Contents

Chapter 1

The System N-1

The system chosen initially for the reengineering is a popular python framework **“Flask”** developed by Armin Ronacher of Pocoo. Currently, Flask is being developed and supported by the Pallets organisation. Flask is an open source micro web framework and is classified as microframework because it does not require any libraries or tools.

1.1 Purpose of System N1

Flask is a lightweight *WSGI (Web Server Gateway Interface)* web application framework. It is a simple wrapper around *Werkzeug (which helps in the request, response and utility functions)* and *Jinja (handles templates and sandbox)*. It does not enforce any dependencies or project layouts but it offers flexibility to the developers to use the tools and libraries to add the required functionalities.

The flask project has been developed in **Python** and supports HTML and JSON as well which makes it easy to integrate the front end technologies in the project. The framework is scalable which makes it easier to use for web development in Python.

Open-Source Code Repository https://github.com/pallets/flask

1.2 Rationale for selecting System N-1

Flask has become quite famous among the developers because of its wide application as well as simplicity of the framework. The technical reasons for selecting Flask for reengineering are:

1.2.1 Technical Reasons

- Documentation: The framework is being maintained by Pallets organisation and well documented. The GitHub repository is publicly available and its README.md explains about the Flask and has a small example for it as well.
- Flexible: Flask supports many tools and libraries that can be added to enhance the functionality of the application. It is simpler to add different functionalities and the project structure does not collapse even after adding them.

1.3 Rationale for rejecting system N-1

- Source code smells : The smells found were of low criticality, considering Flask is an operational and well maintained python framework, with an active community.
- Error types: The code smells lack variety. On running code analysis tools on the repository, only one kind of error was found in the implementation of the test files for Cookie support.

Chapter 2

The System N-2

The GitHub Repository chosen here is developed by Chabedalam and is a web application for exam seating arrangement system developed using two java frameworks jsp and servlet.

2.1 Purpose of System N-2

N-2 uses JSP and Servlet for implementing the system and it is aimed to provide seating arrangement to the students. There are various options available in this system like manage rooms, manage floors, manage seating plan. The system aims to automate the process of assigning seats to candidates for exams. It helps optimize the seating arrangement by considering various factors such as the number of candidates, available seats, and any specific requirements or constraints.

By automating the seating arrangement process, the system saves valuable time for exam administrators. It eliminates the need for manual seat allocation, reducing human errors and providing convenience in managing large numbers of candidates. **Open-Source Code Repository** **https://github.com/chabedalam11/Exam-Seating-Arrangement-System-Using-JSP-Servlet**

2.2 Rationale for selecting System N-2

- Scalability and Performance: The System N-2 uses Java which is known for its scalability and performance capabilities. Java can handle multiple concurrent requests efficiently, making it suitable for systems that need to accommodate a large number of users or high traffic volumes during exam periods.

2.3 Rationale for rejecting System N-2

- Security Features: N-2 does not have proper authentication, authorization, and data encryption and any hashing techniques for sensitive information. Thus the system is redundant and reengineering the system does not bring value.
- Overhead: This system uses the frameworks which are not frequently used nowadays which can introduce additional overhead compared to lighter web frameworks or technologies like spring. If the system has specific performance requirements or needs to handle many concurrent users, then there might need to consider alternative technologies that offer better performance characteristics.
- Source Code: System contains more than 2 thousand source lines of code, with semantic issues in its source code accounting for more than half of the smells.
- Testing: The system also lacks a comprehensive testing strategy, including unit tests, integration tests, and system tests. This makes it challenging to identify and fix bugs or introduce new features without breaking existing functionality.

Chapter 3

The System N-3

The GitHub repository chosen as a candidate for reengineering is SearchMate which was designed and developed by KingK619. It is a Command Line Interface tool which helps to search the content of the document over the internet.

3.1 Purpose of System N-3

SearchMate is designed using the Python programming language and it is a CLI tool. It can search the content of a document on the internet and provides the list of URL and also their finding score.

It may also fetch results which are might not be highly ranked up on Google's Search Engine, so one might obtain some rare findings. Moreover, the design of this tool is colorful so that one can understand things properly. The different file types accepted by the system are .txt, .docx and .pdf.

The find score feature helps one to find out the amount of text matching the document. The results can be saved in a .txt file for future reference. The output file generated is named as results.txt and it can be stored anywhere on the computer as per the users choice. **Open-Source Code Repository https://github.com/KingK619/SearchMate**

3.2 Rationale for selecting System N-3

- Working System: The candidate N-3 is a fully functional system written in the Python Language, with negligible inconsistencies. The entire system is run on command line interface which makes it easy to use, and demonstrates good version control.
- Documentation: The system is designed by KingK619 and the repository is well documented. Each feature of the application is well explained in the README.md file. The source code has comments explaining its implementation details.

3.3 Rationale for rejecting system N-3

- Limited scope: The type of documents currently accepted by the system are only .txt, .pdf and .docx. So, it limits the usability of the system for any other type of document.
- Outdated or Inactive: The repository is outdated, with no recent updates or commits. This could indicate that the software is no longer actively maintained or supported. It also has minimal community support and no developer engagement.
- Small Codebase: The codebase for SearchMate is relatively small (less than 1 thousand source lines of code) and straightforward, with well-defined modules and a good documentation. So, there may be less need for reengineering.

Chapter 4

The Candidate R

The repository chosen as a candidate for Reengineering is the **PacmanDQN** developed by *tychovdo* (*Tycho van der Ouderaa*), is a project that seeks to employ deep reinforcement learning techniques to teach an agent to play the well-known mobile game **Pacman**.

4.1 Purpose of System R

This project demonstrates the implementation of Deep Q-Networks, a Reinforcement Learning algorithm utilizing neural networks to develop an AI agent that can navigate the Pac-Man maze, collect pellets, avoid ghosts, and achieve high scores by playing the game autonomously. It shows that the Deep Q-Learning algorithms used in Playing Atari [3] (a task aimed at training AI Agents to play the Atari 2600 games), can be extrapolated to learn the Pacman game. The interactions with the game environment help the agent update its Q-values to make increasingly better decisions over time.

The project is built using the Python Programming Language and employs Google's deep learning framework 'tensorflow', which provides extensive support in python. The candidate system R, develops a simulation environment for the game and incorporates the necessary elements to train the AI agent by building different sized layouts for the pacman to trace.

4.2 Open-Source Code Repository

The Source code for R, the candidate for Reengineering is https://github.com/tychovdo/PacmanDQN

4.3 Rationale for selecting System R

PACMAN has received decades of significant recognition and popularity. The nostalgia and familiarity of the game, along with the simple concepts appeals to a wide audience and its increasing difficulty lead to increasing engagement in the game. Some of the Technical and Non-Technical reasons for choosing it as a candidate for reengineering are listed as follows.

4.3.1 Technical Reasons

- Python-based and Open Source: The source code for the system is developed with Python, due to its shallow learning curve, high readability and legibility, the understanding of the solution domain is enhanced, thereby making it communicable across the team.
- Documentation: The Problem domain of the system is extremely well documented along with support for understanding the underlying algorithm for its implementation. The GitHub repository is well maintained with an appropriate README.md file elaborating the architecture of the system.

- Working System: R is a working system, with minor inconsistencies related to versioning of the system's dependencies, making it an ideal candidate for reengineering.

- Source Code Quality Improvement: The system R, has more than 1.5 thousand source lines of code, in multiple modules with varying complexity. The source code contains numerous pragmatic and semantic issues, which need to be improved to increase its maintainability. These perfectly fits the constraints of the requirements of System R for reengineering.

- Modularity and Scalability: The project leaves a large room for improvement by increasing the cohesion of the source code and reducing its coupling, which makes it easier to understand, test, and extend. This allows for the addition of new features, enhancements, and adaptations to different game environments, making the system more flexible and future-proof.

4.3.2 Non-Technical Reasons

- Entertaining: Pac-Man is an engaging game for players of all ages. It is also a pleasant and addictive one. The project provides entertainment value by exhibiting a Pac-Man AI agent that can play the game independently, maybe scoring highly, and showcasing the abilities of AI in gaming.

- Educational Value: In the Artificial Intelligence era, the use of Machine and Deep Learning models have become increasingly popular. The system implements an extremely crucial reinforcement learning algorithm, Deep Q-Networks. This provides an opportunity for the entire team to become familiar with the architecture of the models and get hands-on experience working on a project.

Bibliography

[1] The Roles and Responsibilites of the Team Members https://docs.google.com/spreadsheets/d/1NjXsd57b4kHZClkviDmL006pi_-imlJnCub5SCKrUt0/edit?usp=sharing

[2] The GitHub Link of Project https://github.com/roshansadath/SOEN6431-SCM-Project

[3] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, Martin Riedmiller, (2013, December 19). *Playing Atari with Deep Reinforcement Learning*, http://arxiv.org/abs/1312.5602.

[4] Flask - Armin Ronacher https://github.com/pallets/flask

[5] Exam Seating Arrangement System - Chebadalam https://github.com/chabedalam11/Exam-Seating-Arrangement-System-Using-JSP-Servlet

[6] SearchMate - KingK619 https://github.com/KingK619/SearchMate

[7] Deep Reinforcement Learning based PacMan Hack - tychovdo https://github.com/tychovdo/PacmanDQN.