DEPARTMENT OF COMPUTER SCIENCE

SOFTWARE ENGINEERING PROJECT

Mentored By: **Dr. SANGEETA SHARMA**

YouPlay

Documentation

Anuj A. Gupta
151112096
CSE -1

Index

I.	Software engineering and their phases	3
II.	Problem Statement & detailed description	8
III.	Software Requirements	10
IV.	Data Flow Diagrams (DFD)	11
V.	Class Diagram	13
VI.	SRS	14

a. Introduction

b. Purpose

c. Scope

d. Overview

e. Product Perspective

Software Engineering & Its Phases

Software is instructions (computer programs) that when executed provide desired function and performance. Software are data structures that enable the programs to adequately manipulate information. Software are documents that describe the operation and use of the programs.

The work associated with software engineering can be categorized into **three generic phases**, regardless of application area, project size, or complexity.

The **Definition Phase** focuses on *what.* That is, during definition, the software engineer attempts to identify what information is to be processed, what function and performance are desired, what system behavior can be expected, what interfaces are to be established, what design constraints exist, and what validation criteria are required to define a successful system. The key requirements of the system and the software are identified. Although the methods applied during the definition phase will vary depending on the software engineering paradigm (or combination of paradigms) that is applied, three major tasks will occur in some form: system or information engineering , software project planning, and requirements analysis.

The **Development Phase** focuses on *how*. That is, during development a software
engineer attempts to define how data are to be structured, how function is to be implemented within a software architecture, how procedural details are to be implemented, how interfaces are to be characterized, how the design will be translated into a programming

language (or nonprocedural language), and how testing will be performed. The methods applied during the development phase will vary, but three specific technical tasks should always occur: software design, code generation, and software testing.

The **<u>Support Phase</u>** focuses on *change* associated with error correction, adaptations required as the software's environment evolves, and changes due to enhancements brought about by changing customer requirements. The support phase reapplies the steps of the definition and development phases but does so in the context of existing software. Four types of change are encountered during the support phase:
Correction: Even with the best quality assurance activities, it is likely that the
Customer will uncover defects in the software. *Corrective maintenance* changes
the software to correct defects.
Adaptation: Over time, the original environment (e.g., CPU, operating system,
Business rules, external product characteristics) for which the software was developed is likely to change. *Adaptive maintenance* results in modification to
the software to accommodate changes to its external environment.
Enhancement: As software is used, the customer/user will recognize additional
functions that will provide benefit. *Perfective maintenance* extends the software beyond its original functional requirements.
Prevention: Computer software deteriorates due to change, and because of
this, *preventive maintenance,* often called *software reengineering,* must be conducted to enable the software to serve the needs of its end users. In essence,

preventive maintenance makes changes to computer programs so that they can
be more easily corrected, adapted, and enhanced.

A development process consists of various phases, each phase ending with a defined output. The phases are performed in an order specified by the process model being followed. The main reason for having a phased process is that it breaks the problem of developing software into successfully performing a set of phase, each handling a different concern of software development.

Software Development Life Cycle, SDLC for short, is a well-defined, structured sequence of stages in software engineering to develop the intended software product.

Basic step

This is the first step where the user initiates the request for a desired software product. He contacts the service provider and tries to negotiate the terms. He submits his request to the service providing organization in writing.

Requirement Gathering

This step onwards the software development team works to carry on the project. The team holds discussions with various stakeholders from problem domain and tries to bring out as much information as possible on their requirements. The requirements are contemplated and segregated into user requirements, system requirements and functional requirements.

Software Design

Next step is to bring down whole knowledge of requirements and analysis on the desk and design the software product. The inputs from users and information gathered in requirement gathering phase are the inputs of this step. The output of this step comes in the form of two designs; logical design and physical design. Engineers produce meta-data and data dictionaries, logical diagrams, data-flow diagrams and in some cases pseudo codes.

Coding

This step is also known as programming phase. The implementation of software design starts in terms of writing program code in the suitable programming language and developing error-free executable programs efficiently.

Testing

An estimate says that 50% of whole software development process should be tested. Errors may ruin the software from critical level to its own removal. Software testing is done while coding by the developers and thorough testing is conducted by testing experts at various levels of code such as module testing, program testing, product testing, in-house testing and testing the product at user's end. Early discovery of errors and their remedy is the key to reliable software.

Operation and Maintenance

This phase confirms the software operation in terms of more efficiency and less errors. If required, the users are trained on, or aided with the documentation on how to operate the software and how to keep the software operational. The software is maintained timely by updating the code according to the changes taking place in user end environment or technology. This phase may face challenges from hidden bugs and real-world unidentified problems.

Problem Statement & Description

"As the number of videos in a YouTube Playlist increases, downloading the whole playlist becomes cumbersome."

What does this mean?

YouTube is the most popular website for video sharing and is used by a wide number of individuals and media corporations .YouTube playlists come in handy while learning certain topic or concept. Videos are arranged according to the concept and are really helpful.

To download videos from YouTube, we have to either use a website as a plugin or a video grabber. Even though downloading a large number of videos is quite cumbersome as videos need to be manually grabbed. YouTube playlists come in handy while learning certain topic or concept. Videos are arranged according to the concept and are really helpful.

How can this problem be solved?

The answer is simple, by using a download manager. A download manager is a computer program dedicated to the task of downloading possibly unrelated stand-alone files from the Internet for storage.

How can such a Download Manager Developed?

This can be accomplished by developing a web-crawler and using the crawled links to download the videos. Python provides a large number of packages for this task. A web crawler can be build which accepts the watch URL of the first video of playlist and the number of videos of the playlist to be downloaded. From this link the downloadable link of the video can be generated and the watch URL of next video can be captured. This concept is implemented in YouPlay.

What is YouPlay?

YouPlay is a download manager written in python 3.0.It is designed to be used on windows operating system. It is a sort of web-crawler cum downloader. It can be used to download a stand-alone video to the whole playlist. It can be used to find the number of videos in a playlist to downloading only few of them.

YouPlay asks the user for the URL of playlist. It then shows the number of videos in the playlist and list of videos. User is supposed to enter the video number from which to begin the download & video number of last video. It also asks for the directory location in which the videos are supposed to be saved. It even creates the directory for you if it does not exist.

YouPlay works in the background and notifies the user on completion of the download by ringing an alert tone. YouPlay always downloads the videos in the best file format available.

Software Requirements

Active Internet Connection

As YouPlay is a download manager, it requires an active internet connection.

Python 3.0 Interpreter

YouPlay has been coded in python3.0. Hence it requires python3.0 interpreter (IDLE) for its execution.

Windows Operating System

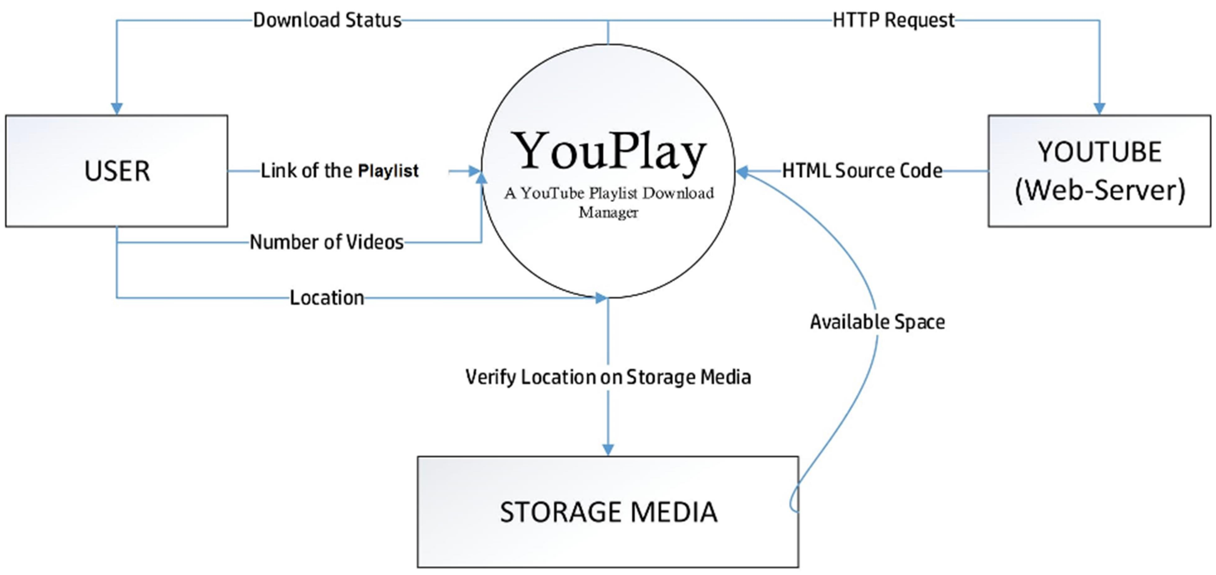
Any version of windows operating system is required for the software.

Space on Storage Media

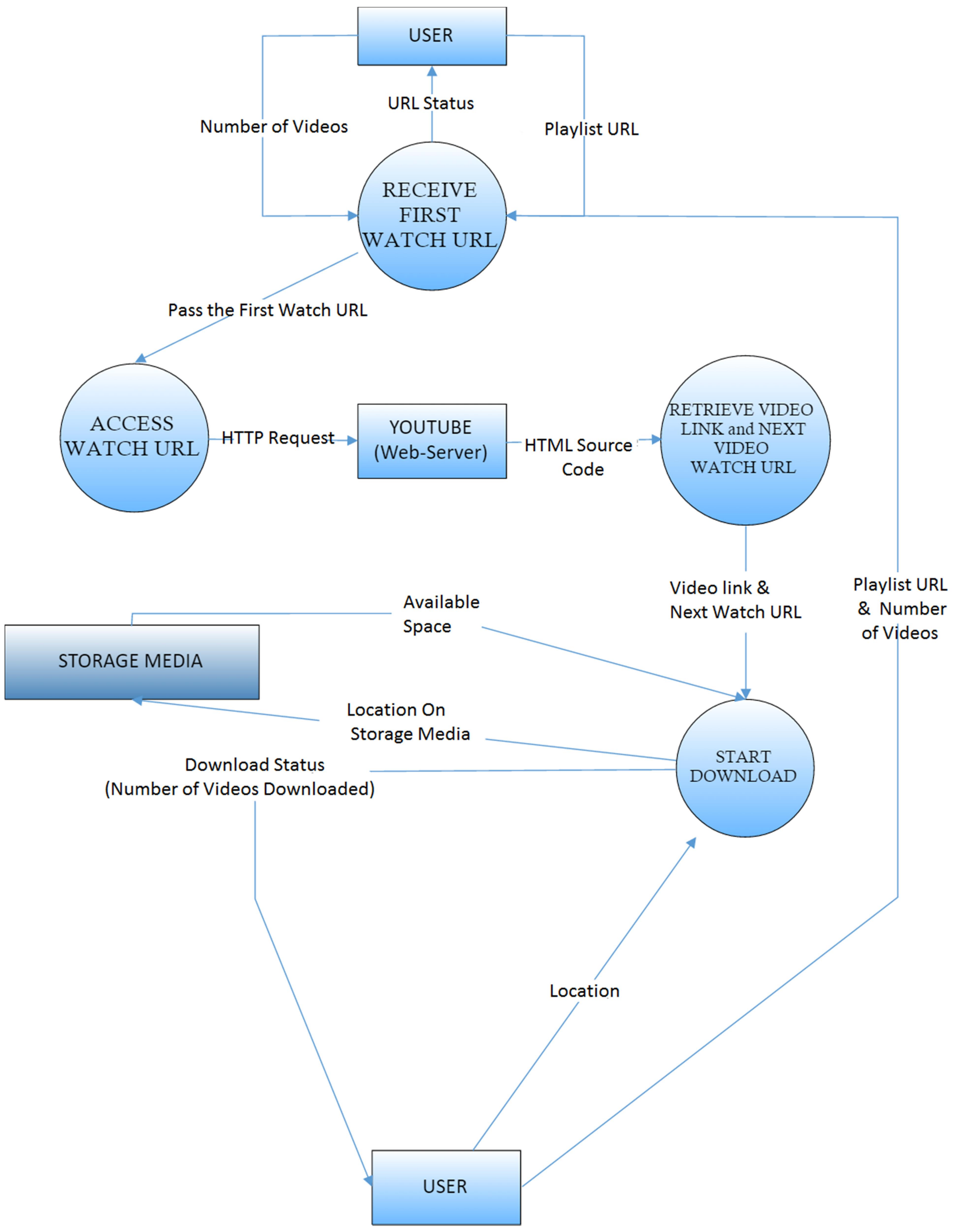
YouPlay requires at least 100Mb of space for its execution as the size of videos differ from playlist to playlist.

Data Flow Diagram

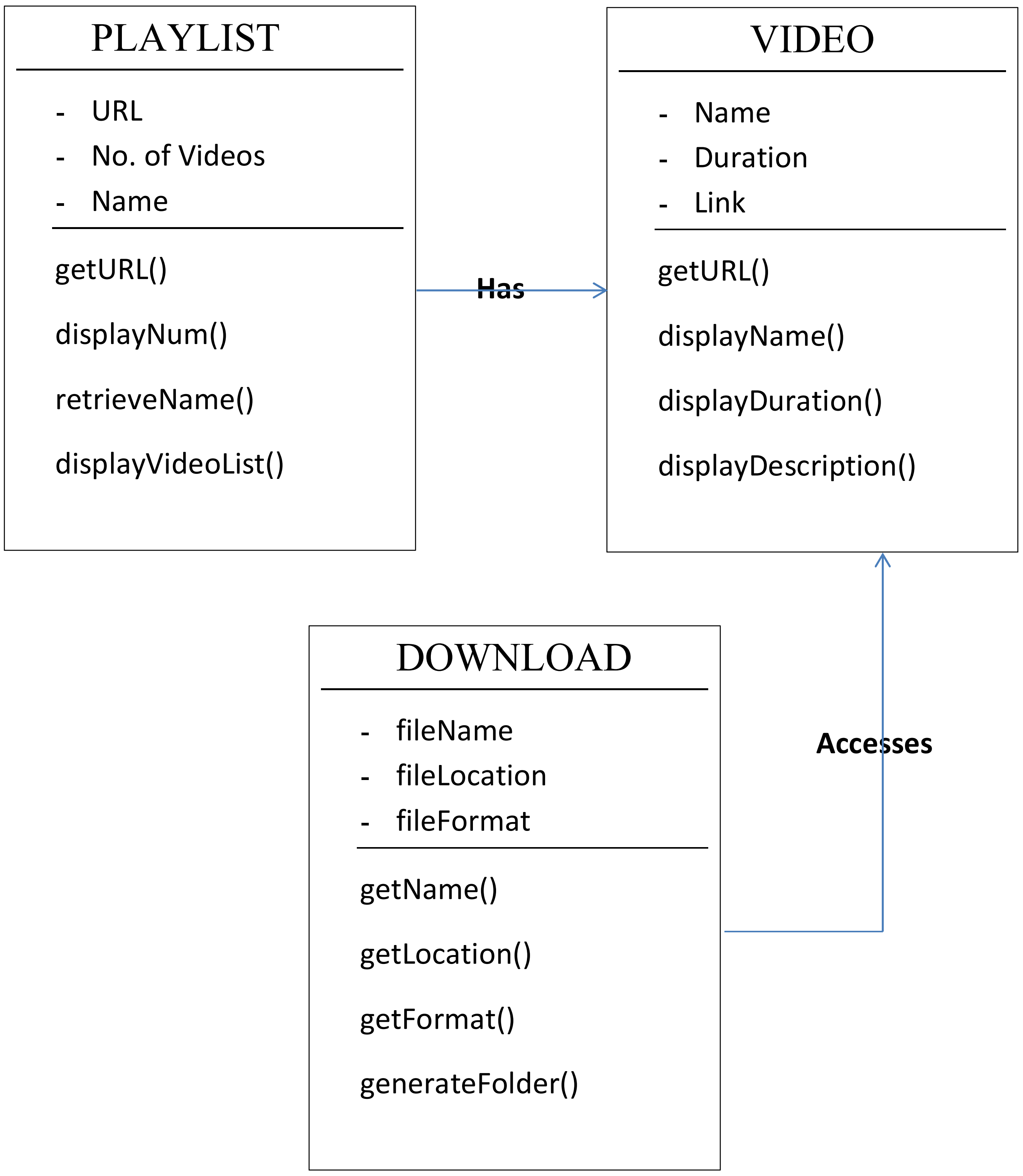
Context Free Diagram



Logical Data Flow Diagram



Class Diagram



Software Requirements & Specifications

Introduction

YouPlay is a download manager written in python 3.0.It is designed to be used on windows operating system. It is a sort of web-crawler cum downloader. It can be used to download a stand-alone video to the whole playlist. It can be used to find the number of videos in a playlist to downloading only few of them.

Purpose

The purpose of YouPlay is to download a YouTube playlist for the user. It can be used to download a stand-alone video to the whole playlist.

Scope

The scope of the project is the system on which the software is installed, i.e. the project is developed as a desktop application, and it will work for a particular user.

Overview

The purpose this document is to present a detailed description of the YouPlay. It will explain the purpose and features of the software, the interfaces of the software, what the software will do, the constraints under which it must operate and how the software will react to external stimuli. This document is intended for both the end users and the developers of the software.

Product Perspective

The product YouPlay is an independent product and does not depend on any other product or system. The product will automate various tasks associated with handling videos of the playlist and organizing the downloaded videos according to their playlist sequence.