Quality Assurance Plan

for

Time-Table Generator

Version 1.0

CS-08

Indian Institute of Information Technology Vadodara

Team Members

Aman Yadav (201651007)

DakshKumar Gondaliya (201651014)

Kirtika Singhal (201651024)

Mayank Pathela (201651029)

Nikhil Sachan (201651034)

Parmeshwar Kumawat (201651035)

Revision History

Version	Date	Name	Description
1	09/10/2018	Kirtika Singhal	Initial Document

Table of Contents

1. Introduction

1.1 Purpose

The purpose of this Software Quality Assurance Plan (SQAP) is to define the techniques, procedures, and methodologies that will be used for TimeTable Generator to assure timely delivery of the software that meets specified requirements within project resources.

1.2 Scope

The use of this plan will help assure the following: (1) That software development, evaluation and acceptance standards are developed, documented and followed. (2) That the results of software quality reviews and audits will be given to appropriate management. (3) That test results adhere to acceptable standards.

2. Applicability

The processes described in this plan are used during the requirements, design, implementation, and test of the software. TimeTable Generator is an application that aims to provide a possible timetable solution with the minimum number of clashes between slots. It relieves the user of much of the hard work required for generating timetable manually, leaving him with more time to apply the skills and judgment where they are needed in order to produce a timetable of the highest quality.

3. Applicable Documents

The following documents can be used as requirements for the design and manufacture of the software and form a part of this document:

1. Software Requirement Specification
2. Software Project Plan
3. Software Development Plan (SDP)
4. Software Test Plan (STP)
5. Configuration Management Plan

4. Program Management And Planning

4.1 Organisation

Professor: Dr. Asim Banerjee

Teaching Assistants:

1. Akansha Tyagi
2. Rashmi Maheshwari

Developers:

1. Aman Yadav (201651007)
2. Dakshkumar Gondaliya (201651014)
3. Kirtika Singhal (201651024)
4. Mayank Pathela (201651029)
5. Nikhil Sachan (201651034)
6. Parmeshwar Kumawat (201651035)

4.2 Responsibilities

Professor: The Professor will supervise and evaluate the application submitted by the developers.

Teaching Assistants (TAs): Providing guidance to the software developers.

Developers: Since the project is divided into subtasks and each subtask is being developed individually, a developer is responsible for ensuring the quality of the project.

4.3 Tasks

The following tasks will be conducted in order to ensure quality assurance:

1. **Driving Requirements**: The developer should ensure that the software requirement specification document clearly states the functionality of the software and unambiguously declares the requirements that must be satisfied. In addition, descriptions of the scope should clearly outline what the software will allow and not allow.

2. **Design**: The developer and professor will conduct reviews and analyses of the construction of the software. Strengths and weaknesses of various design techniques will be discussed and scrutinized.

3. **Implementation**: Informal code reviews will be conducted by the developer on a regular basis to ensure consistency with the design and the detection of any error. Also, a document will be produced for purpose of maintainability and future work.

4. **Testing**: Developer will conduct tests as presented in the Software Test Plan to ensure the requirement satisfaction and reliability of the software.

5. SQA Program Requirements

This section defines the SQA review, reporting, and auditing procedures used to ensure that internal and external software deliverables are developed in accordance with this plan and contract requirements. Internal deliverables are items that are produced by software development and then filed using configuration control procedures. e.g. requirements specifications, software architecture descriptions, executable code. External deliverables are items that are produced for organizations outside of the organization which is developing that software, e.g. user manuals.

5.1 SQA Program Audits

In order for SQA to evaluate compliance with the SDP, the STP, SQA will review and approve the above plans. These plans will specify that the evidence of work generated is adequate to ensure compliance with project and contract requirements. Audits performed shall include an examination of both internal and external software deliverables.

5.1.1 Scheduled Audits

SQA will generate and maintain an Audit Schedule. Audits will occur at the end of each development phase as indicated in the SMP. The results of audits will be discussed with the individual most responsible for the production of the deliverable.

5.1.2 Unscheduled Audits

SQA will perform random and unannounced audits to ensure the corrective actions agreed to during the Scheduled Audits are being followed. The results of audits will be discussed with the individual most responsible for the production of the deliverable.

5.2 Software Documentation

SQA will review all the deliverable software documentation including software plans. Review checklists will be used to review these documents. These reviews will help ensure that documentation is in compliance with generated plans and procedures.
In general, the essential software documentation should include:

1. Software Requirement Specification (SRS): provides a detailed description of the entire project, the goals of the software, constraints, and requirements for the software to satisfy.
2. Software Project Plan: illustrates the major milestones and provides a rough timeline for the project and estimation of the size and effort of the project.
3. Software Test Plan: provides a description of test cases during testing
4. User Manual: instructions on how to use software
5. Quality Assurance Plan: provides a plan for software quality assurance

6. Project Reviews

6.1 Formal Reviews

At least one week prior to the delivery of documents to the client for a formal review, SQA will review the Document List that is generated by the Software Project Engineer (SPE). This list identifies all the documents and revision that will be submitted for the formal review. SQA will review software related documents identified on this list to ensure that the indicated revision is or will be available in time for shipment to the client.

6.2 Informal Reviews

6.2.1 Design Walk-throughs

SQA will be invited to all design walk-throughs throughout the entire software development life-cycle to ensure that peer reviews of the software design are conducted. The Software Project Manager (SPM) will ensure that a verifiable process is used to identify all action items generated during this review process. SQA will audit this process to ensure that all action items have been addressed.

6.2.2 Code Walk-throughs

SQA will be invited to all code walk-throughs to ensure that peer reviews of the source code are conducted. The SPM will ensure that a verifiable process is used to identify all action items generated during this review process. SQA may then audit this process to ensure that all action items have been addressed.

6.2.3 Baseline Quality Reviews

These reviews will be conducted by SQA prior to any baseline release of executable code. This review ensures that:
(1) the code has been tested and meets module specifications;
(2) that any changes to application software module design documents have been identified;
(3) that appropriate validation tests have been run;
(4) that the functionality of the baseline is documented.
(5) that all software design documentation complies with this plan and other applicable plans and procedures.
(6) that tool and techniques used to produce and validate the Software Sub-System are identified and controlled.

7. Problem Reporting

If any problems are encountered throughout the duration of the project, the software developer can report and discuss the problems with the professor.

8. Tools, Techniques, and Methodologies

For determining whether the software requirements were satisfied, a software test plan will be written. This plan will provide an overview of the methodologies, timetables, and resources for testing the software. Testing will commence in three primary ways:

1. **Unit Test**
   Individual classes will be tested to ensure reliability and functionality within a unit level. Furthermore, the testing module will be created before the tested code to ensure that the code is testable.

2. **Integration Test**

Several classes will be tested together to ensure sufficient execution and compliance with the requirements after integration.

3. System Testing

The whole system shall be used for system testing to ensure all requirements are satisfied, and reliability will be included in the testing to measure successful rate.

Any document that will be updated will have a version number.