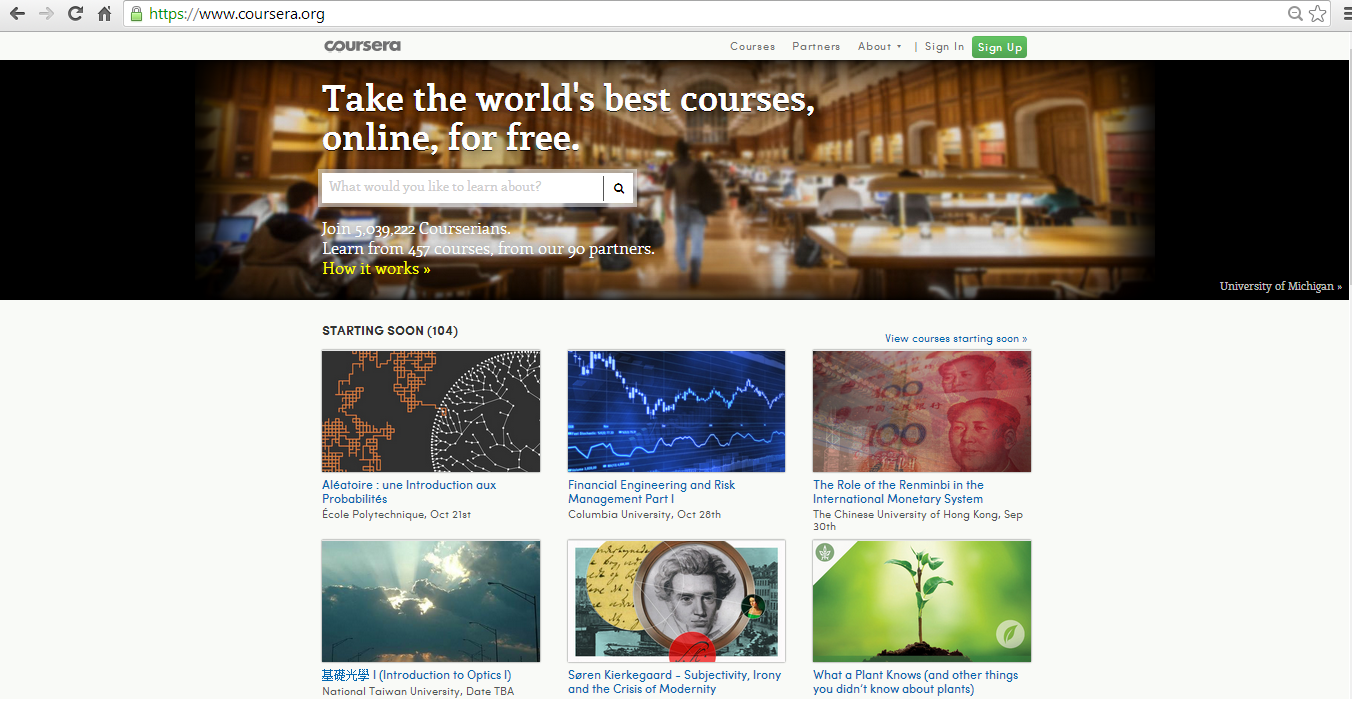
Coursera.org

SRS(Software Requirements Specification)



Coursera is an educational technology company offering massive open online courses founded by computer science professors Andrew Ng and Daphne Koller from Stanford University

Assignment

Software Engineering

**Submitted to:**

Mr.AdityaPancholi

**Submitted by:**

RakeshKumar(31)

RiyaMalhotra(32)

SanjuYadav(33)

**Date of Submission:**

5th October 2013

MODEL USED:

Among the various models, the best suited for this website is EVOLUTIONARY PROTOTYPING MODEL.

We are using this model because many suggestions and new requirements may come from the usersduring the development of the system software. Instead of freezing the requirements before a design or coding can proceed, a prototype is built to understand the requirements.This prototype is developed based on the currently known requirements.When the user is satisfied, the prototype code is brought up to the standards needed for a final product

1. INTRODUCTION

The Coursera is a MOOC (Massive Open Online Course) platform that partners with the top universities and organizations in the world to offer courses online for anyone to take for free.

**1.1. PURPOSE**

The proposed SRS document for Coursera has the following purposes:

* communicate understanding of user and system requirements
* serve as a reference document for designers, developers and testers
* Understanding the overall working of Coursera.

**1.2. SCOPE**

The Coursera platform offers a learning environment vastly different from the traditional

classroom. Coursera’s global community of learners represents not only a huge potential

audience, but also a vast spectrum of ages, language proficiencies, academic and professional

training, and cultural backgrounds. Across the disciplines, Coursera courses appear to share a

number of common considerations when developing learning goals and objectives:

• Concise, specific, and well defined goals and objectives help instructors to teach more

effectively, and help students to better navigate the course, enabling them to make informed

decisions about how to work with the materials, activities and assessments.

• Students might not know what to expect or how to succeed in a massive open online course.

Effective communication between instructor and students is carried out by providing a well defined structure and instructions.

2. OVERVIEW

**2.1. PRODUCT PERSPECTIVE**

The product is supposed to have an online learning platform which reaches to a massive audience. It supports multiple languages. The courses are categorized according to their fields (subjects).If a user is enrolled for a course, it provides video lectures, quizzes and programming exercises.

**2.2. PRODUCT FEATURES**

**2.1.1. COURSE CHARACTERIZATION**

Courses are categorized according to different languages and fields (subjects). User can easily browse through different courses. Courses are also classified according to whether signature track or they are starting soon or not.

**2.2.2. USER PROFILE**

User can make a profile where they can provide information about themselves like their photo, description about themselves, birthdate etc. Students can also set privacy settings.

**2.2.3. SEARCH BAR**

User can directly view a course by specifying the title, university offering the course or the instructor teaching the course.

**2.2.4. WATCHLIST**

User can add a course to watchlist. By adding a course to watchlist, user can keep a watch on the course like when the course is going to start next time, who is teaching the course by email and also on their homepage if they are sign up with coursera.

**2.2.5. VIDEO LECTURES**

Coursera provides video lectures, which can be viewed by user.

**2.2.6. PROGRAMMING EXERCISES**

On the regular interval programming exercises are provided to the user. User can opt for these exercises and submit them for evaluation. This helps users to improvise themselves.

**2.2.7. QUIZZES**

**2.2.8. DISCUSSION FORMS**

**2.2.9. SIGNATURE TRACK**

**2.2.10. COURSERA MEETUP**

**2.2.11. DOWNLOAD LECTURE NOTES AND SLIDES**

**2.2.12. EMAIL FACILITY**

**2.3. USER CHARACTERISTICS**

User: The users who sign up with Coursera and later get themselves enrolled for the courses of their choice.

Instructor (Teacher): The user who host the courses. Only the instructors from recognized university can host courses with prior permission.

**2.4. ASSUMPTION AND DEPENDENCIES**

* It is assumed that all users can access the system through internet.
* All the plugins of the web browser are working on the student machine to access courses.

3. FUNCTIONAL REQUIREMENTS

**3.1. COURSE**

**INTRODUCTION:**

The user can browse through different courses offered by Coursera and know about the starting date and duration of the course.

**INPUT:**None

**PROCESSING:**

After selecting course user is directed to a page where different courses are listed. Users can select a particular language and view the courses offered in that language or user can view courses according to the selected field (subject). User can also view only about the courses which are available for sign track. User can select a particular course at this time to know more about the course.

**OUTPUT:**

If user selected the particular course then directed to the event which displays the information about the courses.

**3.2. SIGN UP**

**INTRODUCTION:**

To enroll for the courses the user need to first sign up with Coursera.

**INPUT:** None

**PROCESSING:**

After selecting the sign up option, sign up form is displayed. User fills the email id, user name and password in the form. After this user account is created and user is directed to his/her homepage and a mail is sent to the user.

**OUTPUT:**

A user homepage is created and user is directed to its homepage.

**3.3. SIGN IN**

**INTRODUCTION:**

After sign up, user can any time visit his/her homepage.

**INPUT:**A user must sign up with Coursera.

**PROCESSING:**

User selects the sign in option. Sign in form is displayed. User fills the email id and password in the form. If email id and password is correct user is directed to its homepage. If user id or password is wrong an error is displayed. If user forgets his/her password, he/she can select the forget password option at this stage and request for a new password. For generating a new password a link is mailed to the user.

**OUTPUT:**

User is sign in his/her account and user homepage is displayed.

**3.4. UPDATE PROFILE INFORMATION**

**3.4.1. COURSE DASHBOARD**

**INTRODUCTION:**

User can view all the courses enrolled by him/her or also about the courses added in his/her watch list.

**INPUT:**

A user must be sign in with Coursera. User must enroll for courses or add them to watch list.

**PROCESSING:**

After selecting course dashboard an event will be generated this gives the information about the courses enrolled by the user.Users can un-roll from any course which he/she had enrolled.

**OUTPUT:**

User course dashboard is displayed.

**3.4.2. SETTING PROFILE**

**INTRODUCTION:**

After sign in user can add more features to his/her account like photo,description, birthdate etc. User can also change the privacy settings.

INPUT: User must sign in with the Coursera.

**PROCESSING:**

After selecting the profile, user profile page is displayed. User edits his/her profile information and can save the changes.

**OUTPUT:**None

**3.4.3. COURSE RECORDS**

**INTRODUCTION:**

User can view information about all the courses completed by him/her. User can also view the grades which he/she scored after the completion of course.

**INPUT:**

User must sign in with Coursera and must enroll with at least one course which is completed.

**PROCESSING:**

User selects the course records options; his/her course record page is displayed.

**OUTPUT:**None

**3.4.4. CAREER SERVICES**

**INTRODUCTION:**

Apply for the jobs to Coursera Career Service partners.

**INPUT:**

User must sign in with Coursera.

**PROCESSING:**

User selects the career services options. User is directed to his/her career service page, where user can upload his/her curriculum vitea, linkedln profile etc. and save the changes.

**OUTPUT:**None

3.4.5. SETTINGS

**INTRODUCTION:**

User can change his/her user name, password, email address etc.

**INPUT:**

User must sign in with Coursera.

**PROCESSING:**

User selects the settings options. User account setting form is displayed. User can change information like user name, password, email address etc. and can save the changes.

**OUTPUT:**None

**3.5. ENROLL FOR THE COURSE**

**INTRODUCTION:**

A user can browse through different courses and enroll for any course of his/her choice.

**INPUT:**

A user must sign in.

**PROCESSING:**

Userbrowses through the different courses and select a course. User is directed to the course description page. User can read the course description and watch the intro video. If user wants to sign up for the course, user selects the enroll event and is directed to course homepage.

**OUTPUT:**

User gets enrolled in the course and gets regular updates about the course by email and at course homepage also.

**3.6. EXERCISES:**

**INTRODUCTION:**

After sign-in and selecting the go to class option for a particular course the student can attempt the exercises available for that course.

**INPUT:**

The student must sign-in Coursera and later sign-up for the desired course and then selects go to class option. Student clicks on the Exercises option.

P**ROCESSING:**

A list of week wise exercises is available for that course till the current week. Students can view information such as due date, hard deadline, effective score and no. of attempts made corresponding to each exercise. Student selects the exercise he/she wants to take by clicking on the corresponding attempt exercise button. Selected exercise is displayed. After attempting, a student can either save the selected answers for later submission or submit it.

**OUTPUT:**

If submit button is clicked, the assignment will be sent for peer assessment.

**3.7. PROGRAMMING ASSIGNMENTS:**

**INTRODUCTION:**

After sign-in and selecting the go to class option for a particular course the student can attempt programming assignments available for that course.

**INPUT:**

The student must sign-in Coursera and later sign-up for the desired course and then selects go to class option. Student clicks on the Programming Assignments option.

**PROCESSING:**

A list of week wise programming assignments is available for that course till the current week. Student can view due date and hard deadline information. Student can view the instructions by clicking on the button available corresponding to the one he/she wishes to take. She/he can view the assignments requirements specification, the checklist of frequently asked questions and hints, web submission and assessment report guidelines. On completion of the assignment, a student can submit it.

**OUTPUT:**

If submit button is clicked, the assignment will be sent for peer assessment.

**3.8. JOB INTERVIEW QUESTIONS:**

**INTRODUCTION:**

After sign-in and selecting the go to class option for a particular course the student can attempt Job interview questions available for that course.

**INPUT:**

The student must sign-in Coursera and later sign-up for the desired course and then selects go to class option. Student clicks on the Job Interview Questions option.

**PROCESSING:**

A list of week wise job interview questions is available for that course till the current week. Student can view information such as due date, hard deadline, effective score and no. of attempts made corresponding to each week’s questions. Student can view the questions he/she wants to solve by clicking on the corresponding button. Selected question set is displayed. After attempting, a student can either save the selected answers for later submission or submit it.

**OUTPUT:**

If a student clicks on the Submit Answers button, he/she will get a hint. The answers will not be assessed.

**3.9. LECTURES:**

**INTRODUCTION:**

After signing in and selecting the go to class option for a particular course the student can watch the videos available for the particular course.

**INPUT:**

The student must sign in Coursera and later sign-up for the course. Student clicks on the Lectures option.

**PROCESSING:**

A list of week wise lectures is available for that course till the current week. Student can view the lecture in video (MP4), slides, subtitles (txt, srt) format by clicking on the corresponding button.

OUTPUT: On the button click, the corresponding lecture is displayed.

**3.10. DISCUSSION FORUM:**

**INTRODUCTION:**

After sign-in and selecting the go to class option for a particular course the student can post his/her doubts/comments on the discussion forums.

**INPUT:**

The student must sign in Coursera and later sign up for the course. Student clicks on the discussion forum option.

**PROCESSING:**

The discussion forum page for the course is displayed. Student can post his/her doubt on the discussion forum. He/she can view other members’ doubts, his/her latest activity. Student can even subscribe for the email updates.

**OUTPUT:**

After the click on a specific forum, its sub-forum page is displayed.

**3.11. ERRATA:**

**INTRODUCTION:**

After sign-in and selecting the go to class option for a particular course the student can view the errors in the lecture slides and videos.

**INPUT:**

The student must sign in Coursera and later sign up for the course. Student clicks on the Errata option.

**PROCESSING:**

The errata page is displayed which contains week wise errors till the current week. Student can view these errors.

**OUTPUT:**

None

**3.12. BOOK SITE:**

**INTRODUCTION:**

After sign-in and selecting the go to class option for a particular course the student can view the book sites available for that course.

**INPUT:**

The student must sign in Coursera and later sign up for the course. Student clicks on the booksite option.

**PROCESSING:**

A new page is displayed which contains the information regarding the available book sites for that course. A student can read the material provided by these books.

**OUTPUT:**

None

ID

Password

ADMINISTRATOR

Validate\_course()

Login()

LogOut()

PROFILE

REGISTERED USER

GUEST

Name

Location

Email\_id

Username

Password

Sign\_Up()

View\_course()

Sign\_in()

Sign\_out()

Update\_profile()

Settings()

Signature\_track

Evaluate()

Add\_course()

Propose\_course()

INSTRUCTOR

Instructor\_id

University\_Id

University\_id

Attempt()

Upload()

Exercise\_id

EXCERCISE

Attempt()

Upload()

Assignment\_id

PROGRAMMING ASSIGNMENTS

View()

Upload()

Download()

Lecture\_id

Instructor\_id

VIDEO LECTURES

Enroll()

Uenroll()

View\_coursedashboard()

View()

Course\_name

Course\_id

Instructor

COURSE

ENROLL

**Fig: - CLASS Diagram**

UNENROLL

ENROLL

GO TO CLASS

USER’S

HOME PAGE

SIGN-IN

BROWSE

COURSE

UPDATE

SIGN UP

HOME

PAGE

COURSE

DASHBOARD

PROFILE

VIEW LECTURES

VIEW COURSE

DISCUSS

SETTINGS

PROGRAMMING

EXERCISE

ASSIGNMENT

SIGN-OUT

**Fig: - STATE Diagram**

<<includes>>

<<extends>>

<User>

<<extends>>

<Registered User>

<<extends>>

Propose

<<extends>>

<Instructor>

<Admin>

**Fig: - Use Case Diagram**

**4. NONFUNCTIONAL REQUIREMENTS**

4.1 **Performance Requirements**

Performance: In every action-response of the system, there should be no immediate delays. In

case of opening windows forms, popping of error messages and saving the settings or sessions

the delay should be below 2 seconds. In case of opening databases and computing there should

be no delays and the operation should be performed in less than 2 seconds.

4.2 **Safety Requirements**

**Consistency**: Checking the fact that all clients must be attached to one server, so there is an

appropriate control of the information.

4.3 A**vailability:** Checking that the system always has something to function and always pop

up error messages in case of component failure.

4.4 **Usability:** Checking that the system is easy to handle and navigates in the most expected

way with no delays. In that case the system program reacts accordingly and transverses

quickly between its states.

4.5 **Functionality:** Checking that the system provides the right tools to support all its

features.

**Risk Probability and Impact**

**TECHNOLOGY RISK:**

1. Denial of service leading to customer dissatisfaction.

**Probability:** Low

**Impact:** Serious

**Mitigation:**Before customer use, do proper testing in order to avoid this risk.

2. Hacking (website is defaced due to weak internal controls) leading to unauthorized access to confidential data.

**Probability:** Low

**Impact:** Catastrophic

**Mitigation:** Apply secure permissions (to avoid) and good internal controls.

**PEOPLE’S RISK:**

1. It is difficult to recruit staff with the skills required within expected time period.

**Probability:** High

**Impact:** Catastrophic

**Mitigation:** Alert customer to potential difficulties and the possibility of delays.

1. Key staff is unavailable at critical times.

**Probability:** Moderate

**Impact:** Serious

**Mitigation:** Reorganize team so that there is more overlap of work and people therefore understand each other’s jobs.

**ORGANIZATIONAL RISK:**

1. The organization is restructured so that different management is responsible for the project.

**Probability:** Moderate

**Impact:** Serious

**Mitigation:** Prepare a briefing document for senior management showing how the project is making a very important contribution to the goals of the business.

1. Organizational financial problems force reductions in the project budget.

**Probability:** Low

**Impact:** Catastrophic

**Mitigation:** Prepare a briefing document for senior management showing how the project is making a very important contribution to the goals of the business and presenting reasons why cuts to the project budget would not be cost-effective.

**REQUIREMENT RISK:**

1. Changes to requirements that require major design rework are proposed.

**Probability:** Moderate

**Impact:** Serious

**Mitigation:** Derive traceability information to assess requirements change impact.

1. Customers fail to understand the impact of requirements changes.

**Probability:** Moderate

**Impact:** Tolerable

**Mitigation:**Update the employers regularly about the status and working assumptions. Get Customers feedback periodically.

**ESTIMATION RISK:**

1. The time required to develop the software is underestimated.

**Probability:** High

**Impact:** Serious

**Mitigation:**investigate using the software tools to get a real time constraint.

**PLANNING AND MANAGEMENT**

**Estimation Using Function Points**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| TRANSCTION FUNCTION | FIELDS/FILES INVOLVED | FTR | DTE | COMPLEXITY | UFP | LANGUAGE |
| Homepage  (EQ) | Fields-search bar, course starting soon, recently added courses, sign in, sign up  Files-Course | 1 | 5 | Low | 3 | HTML/javascript |
| Users home page  (EQ) | Fields-your courses, upcoming courses, watchlist, go to class, search bar  Fields-Course, Profile | 2 | 5 | Average | 4 | HTML/javascript |
| Sign-up(EI) | Fields- name, email-id, password, confirm password, terms of services, submit/validate  File-Profile | 1 | 6 | Low | 3 | PHP |
| Sign-in(EO) | Fields- email-id, password, forget password, submit  File-Profile | 1 | 4 | Low | 4 | PHP |
| Browse Courses(EQ) | Fields- search bar , 24 category options, 12 language options, signature track, starting soon/newest courses, university  File- Course | 1 | 40 | Average | 4 | PHP |
| View course dashboard(EQ) | Fields- Course info, watchlist, recommendations, un-enrol, go to class  File- Course | 1 | 5 | Low | 3 | HTML/javascript |
| Add/Update user profile information(2EI) | Fields- view profile, edit profile, my courses, Privacy, photo, location, gender, birthday(month, day ,year), about me, websites(6) , save changes  File- Profile | 1 | 18 | 2\*Average | 8 | PHP |
| Add/Modify Account Settings(2EI) | Fields- Full name, change name, time zone, change time zone, locale, change locale, email address, change email address, old password, new password, confirm new password, change password  File- Profile | 1 | 13 | 2\*Low | 6 | PHP |
| View Lectures (EO) | Fields- week no., subcategory, topic name, format  File- Course | 1 | 4 | Low | 4 | PHP |
| Attempt exercises (EI) | Fields- week no., subcategory, question  No., submit, save answers  File- Course | 1 | 5 | Low | 3 | PHP |
| View Assignment Instructions (EO) | Fields- specification, checklist, web submissions, assessment report, week no.,  Topic  File- Course | 1 | 6 | Low | 4 | PHP |
| Attempt job interview questions (EI) | Fields- week no., topic, question no., submit, save answers  File- Course | 1 | 5 | Low | 3 | PHP |
| Add or modify course  (2\*EI) | Fields-Course number, course title, course description, start date,  Subject, signature track is available or not, language, submit, confirm or error message  Files-Course1 | 1 | 9 | Low | 6 | PHP |
| Upload lectures  (EI) | Fields-Course number, Course name, lecture id, lecture title, category, duration, format, week, submit, confirm or error message  Files-Course | 1 | 10 | Low | 3 | PHP |
| Upload programming assignments  (External Input) | Fields-Course number, Course name, programming\_ id, programming\_exercise title, due\_date, hard\_deadline, instructions, submit, confirm or error message  Files-Course | 1 | 8 | Low | 3 | PHP |
| Upload exercise  (External Input) | Fields-Course number, Course name, exercise\_id, due\_date, hard\_deadline, submit, confirm or error message  Files-Course | 1 | 7 | Low | 3 | PHP |
| Upload study material  (External Input) | Fields-course number, course name, format, category, submit, confirm or error message  Files-Course | 1 | 6 | Low | 3 | PHP |
| Sign\_in(Administrator)  (EO) | Fields-username, password, submit  Files-Instructor | 1 | 3 | Low | 4 | PHP |
| Course -info(Administration)(EO) | Fields-course\_name,course\_number, start date, duration  Files-Course,Instructor | 2 | 4 | Low | 4 | PHP |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| DATA TRANCTION | FIELDS/FILES INVOLVED | RET | DET | COMPLEXITY | UFP | LANGUAGE |
| User profile | Fields-ID, name, date of birth, location, username, password, email address, display\_photo, gender, courses enrolled | 2 | 10 | Low | 7 | SQL |
| Instructor | Fields-ID, name, date of birth, location, username, password, email address, display\_photo, gender, university | 1 | 10 | Low | 7 | SQL |
| Course | Fields-course name, course number, instructor, subject, start date, duration, university, description, language, signature track | 4 | 10 | Low | 7 | SQL |
|  |  |  |  |  |  |  |

**UNADJUSTED FUNCTION POINTS**

HTML/Javascript

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Function Type | Low | Average | High | FP count |
| EI | 0 | 0 | 0 | 0 |
| EO | 0 | 0 | 0 | 0 |
| ILF | 0 | 0 | 0 | 0 |
| EQ | 2\*3=6 | 1\*4=4 | 0\*6=0 | 10 |
| Total |  |  |  | 10 |
|  |  |  |  |  |

PHP

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Function Type | Low | Average | High | FP count |
| EI | 9\*3=27 | 4\*4=16 | 0\*6=0 | 43 |
| EO | 2\*4=8 | 1\*5=5 | 0\*7=0 | 13 |
| ILF | 0 | 0 | 0 | 0 |
| EQ | 3\*3=9 | 0\*4=0 | 0\*6=0 | 9 |
| Total |  |  |  | 65 |
|  |  |  |  |  |

SQL

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Function Type | Low | Average | High | FP count |
| EI | 0 | 0 | 0 | 0 |
| EO | 0 | 0 | 0 | 0 |
| ILF | 3\*7=21 | 0\*10=0 | 0\*15=0 | 21 |
| EQ | 0 | 0 | 0 | 0 |
| Total |  |  |  | 21 |
|  |  |  |  |  |

**VALUE ADJUSTMENT FATCOR**

|  |  |
| --- | --- |
| GENERAL SYSTEM CHARACTERISTIC | DEGREE OF INFLUENCE |
| Distributed data processing | 3 |
| On-Line data entry | 5 |
| End-user efficiency | 4 |
| On-Line update | 4 |
| Complex processing | 3 |
| Reusability | 3 |
| Installation ease | 3 |
| Operational ease | 4 |
| Multiple sites | 3 |
| Facilitate change | 3 |
| Transaction rate | 3 |
| Heavily used configuration | 3 |
| Performance | 5 |
| Data processing | 4 |

Total degree of influence=50

VAF=(TDI\*0.01)+0.65

VAF=(50\*0.01)+0.65

VAF=1.15

**Function Points**

Function Points = UAF \* VAF

HTML / Javascript

10 \* 1.15 = 11.50 FP

PHP

65 \* 1.15 =74.75 FP

SQL

21 \* 1.15 =24.15 FP

**Source Lines of Code**

HTML / Javascript

11.50 \* avg(48, 63) =638.25

PHP

74.75 \* 53 =3961.75

SQL

24.15 \* 37 =893.55

Total SLOC = 5493.42

SLOC/thousands=5493.42/1000=5.49342

Effort = 3.0 \*(5.49342)^ 1.12 = 25.42 Person Months

Development Time (D) = 2.5\*(25.42)^0.35 =7.76 Months

People required=Effort/Development time=3.28

|  |  |  |
| --- | --- | --- |
| TASKS | DURATION( days) | |
| I. Foundation | | |
| Identify exercise planning team | 2 | |
| Schedule first planning team conference | 1 | |
| Planning Conference (PC) |  | |
| Develop sign-in sheets | 1 |
| Develop agenda | 1 |
| Develop presentation | 2 |
| Identify stakeholders | 1 |
| Assign responsibilities and due dates for each task | 3 | |
| Requirements Phase |  | |
| Identifying needs and project constraints | 15 |
| Establish product statement | 3 |
| Feasibility Study | 5 |
| II. Design and Development | | |
| Scope |  | |
| Identify design objectives | 3 |
| Identify participants | 2 |
| Documentation |  | |
| Develop Exercise Plan (EXPLAN) | 10 |
| Develop Multi-media Presentation and videos | 15 |
| Iteration 1 75 days | |
| Design Stage | 15 |
| Basic diagram | 9 |
| Advanced diagram | 6 |
| Implementation Stage | 40 |
| Homepage | 5 |
| Sign-in, Signup | 5 |
| Course description page | 27 |
| User account page | 3 |
| Testing Stage | 10 |
| Evaluation Stage | 10 |
| Iteration 2 123 days | |
| Review Requirements | 15 |
| Design Stage | 34 |
| Review & evolve diagrams | 21 |
| Graphic user interface design | 13 |
| Implementation stage | 47 |
| Testing stage | 15 |
| Evaluation stage | 12 |
| Deployment | 1 |
| Evaluation of project | 14 |