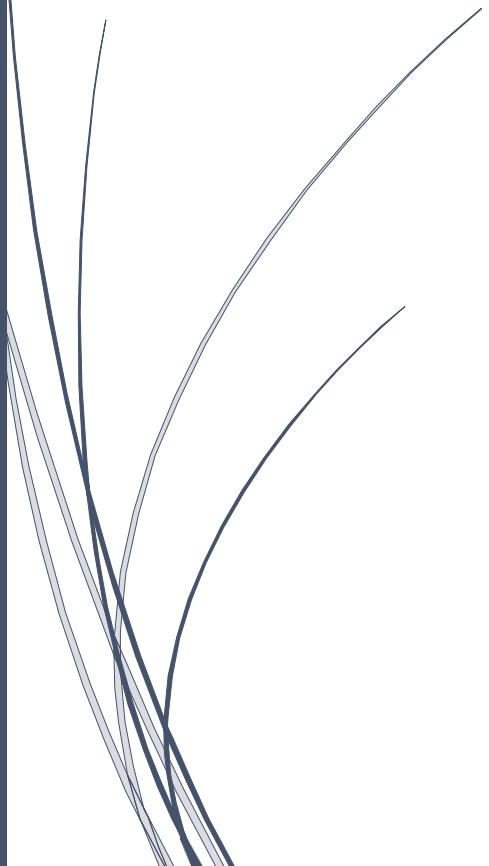




The University of Lahore

Department of Software Engineering



WayForward: Write the Script of the future.



**THE
UNIVERSITY OF
LAHORE**

WayForward: write the script of the future.

FYP-20-29

Hafiz Qasim Qayyum (70069362)

Salman Saleem (70069289)

Moiz Adnan (70069146)

A project submitted in partial fulfillment of the requirement for the degree of

Bachelor of Science in Software Engineering

Supervisor: Mr. Zahid Bashir

**Department of Software Engineering
The University of Lahore, Pakistan**

DECLARATION

We have read the project guidelines and we understand the meaning of academia dishonesty, in particular plagiarism and collusion. We hereby declare that the work we submitted for our final year project, entitled "**WayForward:** write the script of the future".is original work and has not been printed, published or submitted before as research work, publication or any other documentation in any form in any Institute.

Submitted By:

Hafiz Qasim Qayyum

.....

Salman Saleem

.....

Moiz Adnan

.....

Approved By:

Mr. Zahid Bashir

.....

Date of Submission:

.....

ABSTRACT

As we are in the hectic and busy world, the information technologies play a lot in every aspects of human life cycle. So in this project the user mainly going to be the User/Developer community who can benefit from the final product. The main reason behind this production is to make student's life easy by creating such a product call **WayForward:** write the script of the future in order to fulfil their needs in one place, the User/Developer need to get registered with the site to have access and have the benefit of the site also the user can have access to News and updates, Quarries and solution. In conclusion, the aim of this product to give benefits to new and fresh developers.

DEDICATION

Dedicate to our Parents

Dedicated to our respected parents whose utmost love, care and struggle against all odds brought me to this height of knowledge, encouraged me to complete this Degree, and were major driving force behind my all efforts with the blessings of ALMIGHTY ALLAH.

Dedicate to our Teachers

Who teach such a helpful knowledge that helps us to complete our project successfully. The knowledge they taught us will always help us in our Life.

Dedicate to our Friends

Who helps us not only in this project but in our whole life. They stand besides us through every thick and thin.

We are thankful to all of them.

ACKNOWLEDGEMENT

We are thankful to **ALMIGHTY ALLAH** who gave me courage and passion and prayers of my parents and teachers to achieve the goal that was necessary for the degree. Although it was not an easy task, with the useful direction, kind supervision **Mr. Zahid Bashir butt**, it became easy for me to complete the research work. I am grateful to my Project Supervisor because of his profound interest and encouragement throughout the project work. We would also like to thank all the teachers of Software engineering department who helped us a lot during all the semesters we have studied so far and assisted us at becoming good computer scientists. To be specific we would like to thank "**Dr. Mehfooz ul Haq**" for his huge support throughout the eight semesters that we had studied, he always motivated us how to face criticism in a positive way. It was due to their persistent backing that we stand as Software Engineering professionals.

PREFACE

We have made this report file on the Final Year Project. We have tried our best to elucidate all the relevant detail to the project to be included in the report. While in the beginning we have tried to give a general view about the project. In this report, we will provide a comprehensive introduction to the recent developments in the **wayForward:** write the script of the future and also discusses the opportunities and challenges of our system.

Table of Contents

Chapter 1	1
Introduction to the Problem	1
1.1 Introduction	2
1.2 Purpose.....	2
1.3 Objective	2
1.4 Existing Solution	2
1.5 Proposed Solution:.....	2
Chapter2	3
SystemRequirementSpecification	3
2.1 Introduction	4
2.1.1 Purpose.....	4
2.1.2 Scope	4
2.1.3 Reference	4
2.2 OverallDescription	4
2.2.1 Product Perspective	4
2.2.2 Product Functions	6
2.2.3 User Characteristics	7
2.2.4 Constraints.....	7
2.2.5 Assumptions and Dependencies	7
2.2.6 Apportioning of requirements.....	8
2.3 Specific Requirements.....	9
2.3.1 Functional Requirements.....	9
1. Login	9
2. Administrative management system	10
3. Live Editor Management system.....	13
4. Repository management system	14
5. Sales management system.....	16
6. Feedback management system	19
7. Problem solving management system.....	20
8. Code sharing management system	21

9. User management system	22
2.3.2 Non Functional Requirements.....	25
Chapter 3.....	26
UseCaseAnalysis	26
Administrator use case diagram	27
3.1 Administrator Use Case Diagrams & Specification.....	28
3.1.1 UC Login.....	28
3.1.2 UC Forget ID or Password.....	29
3.1.3 UC Register New User	30
3.1.4 UC Approve User Registration	31
3.1.5 UC Block User.....	32
3.1.6 UC Assign Specific Role.....	33
3.1.7 UC Approve User's Queries	34
3.1.8 UC Remove User's Queries.....	35
3.1.9 UC Approve Developer' Script	36
3.1.10 UC Write Code on Live Editor	37
3.1.11 UC Save Code on Local Machine	38
3.1.12 UC Allow Registered User to Save Code In Repository	39
3.1.13 UC Edit User Code	40
3.1.14 UC View User Code Details	41
3.1.15 UC Add Code to Database	42
3.1.16 UC View Code on Database	43
3.1.17 UC Delete Code	44
3.1.18 UC Payment Gateway Integration.....	45
3.1.19 UC Sales and Purchase	46
3.1.20 UC Multiple Currency	47
3.1.21 UC Developer Sales Code	48
3.1.22 UC User Purchase Code	49
3.1.23 UC Payment Report.....	50
3.1.24 UC Review on Codes.....	51
3.1.25 UC Ratings.....	52
3.1.26 UC Ask Query	53
3.1.27 UC Solution of Query.....	54

3.1.28 UC Admin Upload Codes	55
3.1.29 UC User Upload Codes	56
3.1.30 UC User Rate Script	57
3.1.31 UC Allow User to Switch Role	58
3.1.32 UC User Participate In Voting	59
3.1.33 UC Upload Script	60
3.1.34 UC Delete Script	61
3.1.35 UC Update Script	62
3.1.36 UC User Ask Query	63
User Use Case Diagram	64
3.2 User Use Case Diagrams & Specification	65
3.2.1 UC Login	65
3.2.2 UC Forget ID or Password	66
3.2.3 UC Write Code on Live Editor	67
3.2.4 UC Save Code on Local Machine	68
3.2.5 UC Allow Registered User to Save Code In Repository	69
3.2.6 UC Payment Gateway Integration	70
3.2.7 UC Sales and Purchase	71
3.2.8 UC Multiple Currency	72
3.2.9 UC User Purchase Code	73
3.2.10 UC Review on Codes	74
3.2.11 UC Ratings	75
3.2.12 UC Ask Query	76
3.2.13 UC Solution of Query	77
3.2.14 UC User Upload Codes	78
3.2.15 UC User Rate Script	79
3.2.16 UC Allow User To Switch Role	80
3.2.17 UC User Participate In Voting	81
3.2.18 UC Upload Script	82
3.2.19 UC Delete Script	83
3.2.20 UC Update Script	84
3.2.21 UC User Ask Query	85
Chapter4	86

Design.....	86
4.1 Architecture Diagram.....	87
4.2 Entity Relationship Diagram with Data Dictionary.....	88
Data Dictionary against ERD Diagram.....	89
4.3 Data Flow Diagram.....	92
4.3.1 Level 0.....	92
4.3.1 Level 1.....	93
4.4 Class Diagram.....	94
4.5 Activity Diagram.....	95
Admin Panel	95
4.5.1 Login.....	95
4.5.2 Forget ID or Password	96
4.5.3 Register New User.....	97
4.5.4 Approve User Registration	98
4.5.5 Block User	99
4.5.6 Assign Specific Role	100
4.5.7 Approve User's Queries.....	101
4.5.8 Remove User's Queries	102
4.5.9 Approve Developer' Script.....	103
4.5.10 Write Code on Live Editor	104
4.5.11 save Code on Local Machine.....	104
4.5.12 Allow Registered user to Save Code in Repository	105
4.5.13 Edit User Code	106
4.5.14 View User Code Details	106
4.5.15 Add Code to Database.....	107
4.5.16 View Code on Database.....	107
4.5.17 Delete Code	108
4.5.18 Payment Gateway Integration.....	109
4.5.19 Sales and Purchase	109
4.5.20 Multiple Currency	110
4.5.21 Developer Sales Code.....	110
4.5.22 User Purchase Code	111
4.5.23 Payment Report	111

4.5.24 Review on Codes.....	112
4.5.25 Ratings	112
4.5.26 Ask Query.....	113
4.5.27 Solution of Query	113
4.5.28 Admin Upload Codes.....	114
4.5.29 User Upload Codes.....	114
4.5.30 User Rate Script	115
4.5.31 Allow User to Switch Role.....	115
4.5.32 User Participate In Voting.....	116
4.5.33 Upload Script	116
4.5.34 Delete Script	117
4.5.35 Update Script	117
4.5.36 User Ask Query	118
User Panel	119
4.5.37 Login	119
4.5.38 Forget ID or Password	120
4.5.39 Write Code on Live Editor.....	121
4.5.40 save Code on Local Machine.....	121
4.5.41 Allow Registered User to Save Code In Repository.....	122
4.5.42 Sales and Purchase.....	122
4.5.43 Multiple Currency	123
4.5.44 User Purchase Code	123
4.5.45 Payment Gateway Integration.....	124
4.5.46 Review on Codes.....	125
4.5.47 Ratings	125
4.5.48 Ask Query.....	126
4.5.49 Solution of Query	126
4.5.50 User Upload Codes.....	127
4.5.51 User Rate Script	127
4.5.52 Allow User to Switch Role.....	128
4.5.53 User Participate In Voting.....	128
4.5.54 Delete Script	129
4.5.55 Update Script	129

4.6 Sequence Diagram	130
Admin Panel	130
4.6.1 Login	130
4.6.2 Forget ID or Password	130
4.6.3 Register New User.....	130
4.6.4 Approve User Registration	131
4.6.5 Block User	131
4.6.6 Assign Specific Role	131
4.6.7 Approve User's Queries.....	132
4.6.8 Remove User's Queries	132
4.6.9 Approve Developer' Script.....	132
4.6.10 Write Code on Live Editor.....	133
4.6.11 save Code on Local Machine.....	133
4.6.12 Allow Registered user to Save Code in Repository	133
4.6.13 Edit User Code	134
4.6.14 View User Code Details	134
4.6.15 Add Code to Database.....	134
4.6.16 View Code on Database.....	135
4.6.17 Delete Code	135
4.6.18 Payment Gateway Integration.....	135
4.6.19 Sales and Purchase	136
4.6.20 Multiple Currency	136
4.6.21 Developer Sales Code.....	136
4.6.22 User Purchase Code	137
4.6.23 Payment Report	137
4.6.24 Review on Codes.....	137
4.6.25 Ratings	138
4.6.26 Ask Query.....	138
4.6.27 Solution of Query	138
4.6.28 Admin Upload Codes.....	139
4.6.29 User Upload Codes.....	139
4.6.30 User Rate Script	139
4.6.31 Allow User to Switch Role.....	140

4.6.32 User Participate In Voting.....	140
4.6.33 Upload Script	140
4.6.34 Delete Script	141
4.6.35 Update Script	141
4.6.36 User Ask Query	141
User Panel	142
4.6.37 Login	142
4.6.38 Forget ID or Password	142
4.6.39 Write Code on Live Editor.....	142
4.6.40 save Code on Local Machine.....	143
4.6.41 Allow Registered User to Save Code In Repository.....	143
4.6.42 Sales and Purchase.....	143
4.6.43 Multiple Currency	144
4.6.44 User Purchase Code	144
4.6.45 Payment Gateway Integration.....	144
4.6.46 Review on Codes.....	145
4.6.47 Ratings	145
4.6.48 Ask Query.....	145
4.6.49 Solution of Query	146
4.6.50 User Upload Codes.....	146
4.6.51 User Rate Script	146
4.6.52 Allow User to Switch Role.....	147
4.6.53 User Participate In Voting.....	147
4.6.54 Delete Script	147
4.6.55 Update Script	148
4.7 Collaboration Diagram	149
4.7.1 Login.....	149
For Admin.....	149
For User.....	149
4.7.2 Forget ID & Password	149
4.7.3 Register New User.....	149
4.7.4 Approve User Registration	150
4.7.5 Block User	150

4.7.6 Assign Specific Role	150
4.7.7 Approve User's Queries.....	150
4.7.8 Remove User's Queries	151
4.7.9 Approve Developer' Script.....	151
4.7.10 Write Code on Live Editor.....	151
4.7.11 save Code on Local Machine.....	151
4.7.12 Registered User Save Code on Repository	152
4.7.13 Edit User Code	152
4.7.14 View User Code Details	152
4.7.15 Add Code to Database.....	152
4.7.16 View Code on Database.....	153
4.7.17 Delete Code	153
4.7.18 Payment Gateway Integration.....	153
4.7.19 Sales and Purchase	153
4.7.20 Multiple Currency	154
4.7.21 Developer Sales Code.....	154
4.7.22 User Purchase Code	154
4.7.23 Payment Report	154
4.7.24 Review on Code	155
4.7.25 Ratings	155
4.7.26 Ask Query.....	155
4.7.27 Solution of Query	155
4.7.28 Admin Upload Codes.....	156
4.7.29 User Upload Codes.....	156
4.7.30 User Rate Script	156
4.7.31 Switch Roles	156
4.7.32 User Participate in Voting.....	157
4.7.33 Upload Script	157
4.7.34 Delete Script	157
4.7.35 Update Script	157
4.7.36 User Ask Query	158
4.8 State Transition Diagram	159
4.10 Deployment Diagram	160

Table of Tables

Table 1 Functional Requirement Login	9
Table 2 Functional Requirement Forget ID or Password	9
Table 3 Functional Requirement Register New User	10
Table 4 Functional Requirement Approve User Registration	10
Table 5 Functional Requirement Block User	11
Table 6 Functional Requirement Assign specific role	11
Table 7 Functional Requirement Approve user questions and answers	12
Table 8 Functional Requirement Remove user queries	12
Table 9 Functional Requirement Approve developer's scripts	13
Table 10 Functional Requirement Write code on live editor	13
Table 11 Functional Requirement Save code on local machine	13
Table 12 Functional Requirement Allow registered user to save code in repository	14
Table 13 Functional Requirement Edit user code	14
Table 14 Functional Requirement View user code detail	15
Table 15 Functional Requirement Add code to database	15
Table 16 Functional Requirement View code on database	15
Table 17 Functional Requirement Delete code on database	16
Table 18 Functional Requirement Payment gateway integration.....	16
Table 19 Functional Requirement Sale and purchase management	17
Table 20 Functional Requirement Multiple currency option	17
Table 21 Functional Requirement Allow developer to sale code.....	18
Table 22 Functional Requirement Allow users to purchase code	18
Table 23 Functional Requirement Payment report generation.....	19
Table 24 Functional Requirement Review on codes	19
Table 25 Functional Requirement Code rating	20
Table 26 Functional Requirement Ask query	20
Table 27 Functional Requirement Solution of query	20
Table 28 Functional Requirement Admin upload code	21
Table 29 Functional Requirement User upload code	21
Table 30 Functional Requirement Upload script	22
Table 31 Functional Requirement Update script.....	22
Table 32 Functional Requirement Delete script.....	23
Table 33 Functional Requirement Allow user to ask questions	23
Table 34 Functional Requirement Allow user to participate in voting.....	23
Table 35 Functional Requirement Allow user to Switch roles	24
Table 36 Functional Requirement Allow users to rate scripts	24
Table 37 Use Case Specification Login.....	28
Table 38 Use Case Specification Forget ID or Password	29
Table 39 Use Case Specification Register new user	30
Table 40 Use Case Specification Approve User Registration.....	31
Table 41 Use Case Specification Block User	32
Table 42 Use Case Specification Assign Specific Role.....	33
Table 43 Use Case Specification Approve User's Queries.....	34
Table 44 Use Case Specification Remove user's queries	35

Table 45 Use Case Specification Approve Developer' Script	36
Table 46 Use Case Specification Write Code On Live Editor	37
Table 47 Use Case Specification Save Code On Local Machine.....	38
Table 48 Use Case Specification Allow Registered User To Save Code In Repository	39
Table 49 Use Case Specification Edit User Code	40
Table 50 Use Case Specification View User Code Details.....	41
Table 51 Use Case Specification Add Code To Database	42
Table 52 Use Case Specification View Code On Database	43
Table 53 Use Case Specification Delete Code	44
Table 54 Use Case Specification Payment Gateway Integration	45
Table 55 Use Case Specification Sales And Purchase	46
Table 56 Use Case Specification Multiple Currency	47
Table 57 Use Case Specification Developer Sales Code	48
Table 58 Use Case Specification User Purchase Code.....	49
Table 59 Use Case Specification Payment Report	50
Table 60 Use Case Specification Review On Codes	51
Table 61 Use Case Specification Ratings	52
Table 62 Use Case Specification Ask Query	53
Table 63 Use Case Specification Solution of Query	54
Table 64 Use Case Specification Admin Upload Codes	55
Table 65 Use Case Specification User Upload Codes	56
Table 66 Use Case Specification User Rate Script.....	57
Table 67 Use Case Specification Allow User To Switch Role	58
Table 68 Use Case Specification User Participate In Voting.....	59
Table 69 Use Case Specification Upload Script	60
Table 70 Use Case Specification Delete Script.....	61
Table 71 Use Case Specification Update Script.....	62
Table 72 Use Case Specification User Ask Query	63
Table 73 Use Case Specification Login.....	65
Table 74 Use Case Specification Forget ID or Password	66
Table 75 Use Case Specification Write Code On Live Editor	67
Table 76 Use Case Specification Save Code On Local Machine.....	68
Table 77 Use Case Specification Allow Registered User To Save Code In Repository	69
Table 78 Use Case Specification Payment Gateway Integration	70
Table 79 Use Case Specification Sales And Purchase	71
Table 80 Use Case Specification Multiple Currency	72
Table 81 Use Case Specification User Purchase Code.....	73
Table 82 Use Case Specification Review On Codes	74
Table 83 Use Case Specification Ratings	75
Table 84 Use Case Specification Ask Query	76
Table 85 Use Case Specification Solution of Query	77
Table 86 Use Case Specification User Upload Codes	78
Table 87 Use Case Specification User Rate Script.....	79
Table 88 Use Case Specification Allow User To Switch Role	80
Table 89 Use Case Specification User Participate In Voting.....	81
Table 90 Use Case Specification Upload Script	82
Table 91 Use Case Specification Delete Script.....	83

Table 92 Use Case Specification Update Script.....	84
Table 93 Use Case Specification User Ask Query	85
Table 94 Data dictionary against ERD for user	89
Table 95 Data dictionary against ERD for admin	89
Table 96 Data dictionary against ERD for accounts	89
Table 97 Data dictionary against ERD for database.....	89
Table 98 Data dictionary against ERD for save code.....	89
Table 99 Data dictionary against ERD for inventory	89
Table 100 Data dictionary against ERD for sales management	90
Table 101 Data dictionary against ERD for feedback	90
Table 102 Data dictionary against ERD for Q&A	90
Table 103 Data dictionary against ERD for Sell Code.....	90
Table 104 Data dictionary against ERD for Developer	90
Table 105 Data dictionary against ERD for Editor	90
Table 106 Data Dictionary against ERD for code writer.....	91

Table of Figures

Figure 1 Administrator use case diagram	27
Figure 2 Use Case Login.....	28
Figure 3 Use Case Forget ID or Password	29
Figure 4 Use Case Register new user.....	30
Figure 5 Use Case Approve User Registration	31
Figure 6 Use Case Block User.....	32
Figure 7 Use Case Assign Specific Role.....	33
Figure 8 Use Case Approve User's Queries	34
Figure 9 Use Case Remove user's queries	35
Figure 10 Use Case Approve Developer' Script.....	36
Figure 11 Use Case Write Code On Live Editor	37
Figure 12 Use Case Save Code On Local Machine	38
Figure 13 Use Case Allow Registered User To Save Code In Repository.....	39
Figure 14 Use Case Edit User Code.....	40
Figure 15 Use Case View User Code Details	41
Figure 16 Use Case Add Code To Database	42
Figure 17 Use Case View Code On Database	43
Figure 18 Use Case Delete Code	44
Figure 19 Use Case Payment Gateway Integration.....	45
Figure 20 Use Case Sales And Purchase.....	46
Figure 21 Use Case Multiple Currency	47
Figure 22 Use Case Developer Sales Code	48
Figure 23 Use Case User Purchase Code	49
Figure 24 Use Case Payment Report	50
Figure 25 Use Case Review On Codes	51
Figure 26 Use Case Ratings	52
Figure 27 Use Case Ask Query	53

Figure 28 Use Case Solution of Query.....	54
Figure 29 Use Case Admin Upload Codes	55
Figure 30 Use Case User Upload Codes	56
Figure 31 Use Case User Rate Script.....	57
Figure 32 Use Case Allow User To Switch Role.....	58
Figure 33 Use Case User Participate In Voting	59
Figure 34 Use Case Upload Script.....	60
Figure 35 Use Case Delete Script	61
Figure 36 Use Case Update Script	62
Figure 37 Use Case User Ask Query	63
Figure 38 User Use Case Diagram	64
Figure 39 Use Case Login	65
Figure 40 Use Case Forget ID or Password	66
Figure 41 Use Case Write Code On Live Editor	67
Figure 42 Use Case Save Code On Local Machine	68
Figure 43 Use Case Allow Registered User To Save Code In Repository.....	69
Figure 44 Use Case Payment Gateway Integration.....	70
Figure 45 Use Case Sales And Purchase.....	71
Figure 46 Use Case Multiple Currency	72
Figure 47 Use Case User Purchase Code	73
Figure 48 Use Case Review On Codes	74
Figure 49 Use Case Ratings	75
Figure 50 Use Case Ask Query	76
Figure 51 Use Case Solution of Query.....	77
Figure 52 Use Case User Upload Codes	78
Figure 53 Use Case User Rate Script.....	79
Figure 54 Use Case Allow User To Switch Role.....	80
Figure 55 Use Case User Participate In Voting	81
Figure 56 Use Case Upload Script.....	82
Figure 57 Use Case Delete Script	83
Figure 58 Use Case Update Script	84
Figure 59 Use Case User Ask Query	85
Figure 60 Architecture Diagram.....	87
Figure 61 Entity Relationship Diagram with Data Dictionary	88
Figure 62 Data Flow Diagram Level 0	92
Figure 63 Data Flow Diagram Level 1	93
Figure 64 Class Diagram.....	94
Figure 65 Activity Diagram for Login.....	95
Figure 66 Activity Diagram for Forget ID or Password	96
Figure 67 Activity Diagram for Register New User.....	97
Figure 68 Activity Diagram for Approve User Registration	98
Figure 69 Activity Diagram for Block User	99
Figure 70 Activity Diagram for Assign Specific Role	100
Figure 71 Activity Diagram for Approve User's Queries.....	101
Figure 72 Activity Diagram for Remove User's Queries	102
Figure 73 Activity Diagram for Approve Developer' Script	103
Figure 74 Activity Diagram for Write Code on Live Editor.....	104

Figure 75 Activity Diagram for save Code on Local Machine	104
Figure 76 Activity Diagram for Allow Registered user to Save Code in Repository.....	105
Figure 77 Activity Diagram for Edit User Code	106
Figure 78 Activity Diagram for View User Code Details	106
Figure 79 Activity Diagram for Add Code to Database	107
Figure 80 Activity Diagram for View Code on Database.....	107
Figure 81 Activity Diagram for Delete Code.....	108
Figure 82 Activity Diagram for Payment Gateway Integration	109
Figure 83 Activity Diagram for Sales and Purchase	109
Figure 84 Activity Diagram for Multiple Currency	110
Figure 85 Activity Diagram for Developer Sales Code.....	110
Figure 86 Activity Diagram for User Purchase Code.....	111
Figure 87 Activity Diagram for Payment Report	111
Figure 88 Activity Diagram for Review on Codes.....	112
Figure 89 Activity Diagram for Ratings	112
Figure 90 Activity Diagram for Ask Query	113
Figure 91 Activity Diagram for Solution of Query	113
Figure 92 Activity Diagram for Admin Upload Codes	114
Figure 93 Activity Diagram for User Upload Codes	114
Figure 94 Activity Diagram for User Rate Script	115
Figure 95 Activity Diagram for Allow User to Switch Role	115
Figure 96 Activity Diagram for User Participate In Voting	116
Figure 97 Activity Diagram for Upload Script	116
Figure 98 Activity Diagram for Delete Script.....	117
Figure 99 Activity Diagram for Update Script.....	117
Figure 100 Activity Diagram for User Ask Query	118
Figure 101 Activity Diagram for Login.....	119
Figure 102 Activity Diagram for Forget ID or Password	120
Figure 103 Activity Diagram for Write Code on Live Editor	121
Figure 104 Activity Diagram for save Code on Local Machine	121
Figure 105 Activity Diagram for Allow Registered User to Save Code In Repository.....	122
Figure 106 Activity Diagram for Sales and Purchase	122
Figure 107 Activity Diagram for Multiple Currency	123
Figure 108 Activity Diagram for User Purchase Code.....	123
Figure 109 Activity Diagram for Payment Gateway Integration	124
Figure 110 Activity Diagram for Review on Codes	125
Figure 111 Activity Diagram for Ratings	125
Figure 112 Activity Diagram for Ask Query	126
Figure 113 Activity Diagram for Solution of Query	126
Figure 114 Activity Diagram for User Upload Codes	127
Figure 115 Activity Diagram for User Rate Script.....	127
Figure 116 Activity Diagram for Allow User to Switch Role	128
Figure 117 Activity Diagram for User Participate In Voting.....	128
Figure 118 Activity Diagram for Delete Script.....	129
Figure 119 Activity Diagram for Update Script.....	129
Figure 120 Sequence Diagram for Login.....	130
Figure 121 Sequence Diagram for Forget ID or Password.....	130

Figure 122 Sequence Diagram for Register New User.....	130
Figure 123 Sequence Diagram for Approve User Registration	131
Figure 124 Sequence Diagram for Block User	131
Figure 125 Sequence Diagram for Assign Specific Role	131
Figure 126 Sequence Diagram for Approve User's Queries.....	132
Figure 127 Sequence Diagram for Remove User's Queries	132
Figure 128 Sequence Diagram for Approve Developer' Script	132
Figure 129 Sequence Diagram for Write Code on Live Editor.....	133
Figure 130 Sequence Diagram for save Code on Local Machine	133
Figure 131 Sequence Diagram for Allow Registered user to Save Code in Repository.....	133
Figure 132 Sequence Diagram for Edit User Code	134
Figure 133 Sequence Diagram for View User Code Details	134
Figure 134 Sequence Diagram for Add Code to Database	134
Figure 135 Sequence Diagram for View Code on Database.....	135
Figure 136 Sequence Diagram for Delete Code.....	135
Figure 137 Sequence Diagram for Payment Gateway Integration	135
Figure 138 Sequence Diagram for Sales and Purchase	136
Figure 139 Sequence Diagram for Multiple Currency	136
Figure 140 Sequence Diagram for Developer Sales Code.....	136
Figure 141 Sequence Diagram for User Purchase Code.....	137
Figure 142 Sequence Diagram for Payment Report.....	137
Figure 143 Sequence Diagram for Review on Codes.....	137
Figure 144 Sequence Diagram for Ratings	138
Figure 145 Sequence Diagram for Ask Query	138
Figure 146 Sequence Diagram for Solution of Query	138
Figure 147 Sequence Diagram for Admin Upload Codes	139
Figure 148 Sequence Diagram for User Upload Codes	139
Figure 149 Sequence Diagram for User Rate Script	139
Figure 150 Sequence Diagram for Allow User to Switch Role	140
Figure 151 Sequence Diagram for User Participate In Voting	140
Figure 152 Sequence Diagram for Upload Script	140
Figure 153 Sequence Diagram for Delete Script.....	141
Figure 154 Sequence Diagram for Update Script.....	141
Figure 155 Sequence Diagram for User Ask Query	141
Figure 156 Sequence diagram for Login	142
Figure 157 Sequence diagram for Forget ID or Password.....	142
Figure 158 Sequence diagram for Write Code on Live Editor	142
Figure 159 Sequence diagram for save Code on Local Machine	143
Figure 160 Sequence diagram for Allow Registered User to Save Code In Repository	143
Figure 161 Sequence diagram for Sales and Purchase	143
Figure 162 Sequence diagram for Multiple Currency	144
Figure 163 Sequence diagram for User Purchase Code.....	144
Figure 164 Sequence diagram for Payment Gateway Integration	144
Figure 165 Sequence diagram for Review on Codes.....	145
Figure 166 Sequence diagram for Ratings	145
Figure 167 Sequence diagram for Ask Query	145
Figure 168 Sequence diagram for Solution of Query	146

Figure 169 Sequence diagram for User Upload Codes.....	146
Figure 170 Sequence diagram for User Rate Script	146
Figure 171 Sequence diagram for Allow User to Switch Role	147
Figure 172 Sequence diagram for User Participate In Voting	147
Figure 173 Sequence diagram for Delete Script	147
Figure 174 Sequence diagram for Update Script	148
Figure 175 Collaboration Diagram for Admin Login.....	149
Figure 176 Collaboration Diagram for User Login	149
Figure 177 Collaboration Diagram for Forget ID & Password.....	149
Figure 178 Collaboration Diagram for Register New User	149
Figure 179 Collaboration Diagram for Approve User Registration	150
Figure 180 Collaboration Diagram for Block User	150
Figure 181 Collaboration Diagram for Assign Specific Role	150
Figure 182 Collaboration Diagram for Approve User's Queries	150
Figure 183 Collaboration Diagram for Remove User's Queries.....	151
Figure 184 Collaboration Diagram for Approve Developer' Script.....	151
Figure 185 Collaboration Diagram for Write Code on Live Editor	151
Figure 186 Collaboration Diagram for save Code on Local Machine.....	151
Figure 187 Collaboration Diagram for Registered User save Code on Repository.....	152
Figure 188 Collaboration Diagram for Edit User Code.....	152
Figure 189 Collaboration Diagram for View User Code Details	152
Figure 190 Collaboration Diagram for Add Code to Database	152
Figure 191 Collaboration Diagram for View Code on Database	153
Figure 192 Collaboration Diagram for Delete Code	153
Figure 193 Collaboration Diagram for Payment Gateway Integration.....	153
Figure 194 Collaboration Diagram for Sales and Purchase	153
Figure 195 Collaboration Diagram for Multiple Currency	154
Figure 196 Collaboration Diagram for Developer Sales Code	154
Figure 197 Collaboration Diagram for User Purchase Code	154
Figure 198 Collaboration Diagram for Payment Report.....	154
Figure 199 Collaboration Diagram for Review on Code.....	155
Figure 200 Collaboration Diagram for Ratings.....	155
Figure 201 Collaboration Diagram for Ask Query.....	155
Figure 202 Collaboration Diagram for Solution of Query	155
Figure 203 Collaboration Diagram for Admin Upload Codes	156
Figure 204 Collaboration Diagram for User Upload Codes	156
Figure 205 Collaboration Diagram for User Rate Script	156
Figure 206 Collaboration Diagram for Switch Roles	156
Figure 207 Collaboration Diagram for User Participate in Voting	157
Figure 208 Collaboration Diagram for Upload Script	157
Figure 209Collaboration Diagram for Delete Script	157
Figure 210 Collaboration Diagram for Update Script	157
Figure 211 Collaboration Diagram for User Ask Query	158
Figure 212 State Transition Diagram	159
Figure 213 Component Diagram.....	160
Figure 214 Deployment Diagram.....	160

Chapter 1

Introduction to the Problem

1.1 Introduction

The **wayForward** will provide a platform to developers especially fresh developers to write a code and share with others. It's not only a code sharing platform but provides a huge opportunity for sharing ideas, question answers, group discussion, learn technologies and earn money. People can easily find solution for their technical problems like salary calculation, currency conversion and word to pdf converter. It will also facilitate the developers by providing live code editor with drag and drop tags and snippets. This platform will empower the developers and connect them to solutions that enable productivity, growth, and discovery. It is an open platform which allow the people to ask question publicly and privately to find solution for their problems. This platform will also help the software industry and other organization to find and hire skilled developers for their business and providing them the tools they need to share knowledge and work effectively. The **wayForward** will rank the developers with the help of open voting option. In short, this platform will be a complete package for learners, developers, job seekers and organization who needs technical solution for their businesses.

1.2 Purpose

The **wayForward** is an application which target the developers and organization needs digital solution and experts for their business. Developers can help other developers by sharing their code, answers user queries and improve their coding skills on this platform. There are many web applications like **StackOverflow**, **GitHub**, **Codeguru** and **Quora** providing same functionality which we are offering. These applications targeting only single domain but our platform will provide multiple option under one roof.

1.3 Objective

- Code Sharing
- Live code editor with snippets
- Online way of earning for developer
- Developer and code ranking and rating
- Remove distance barriers
- Provide question and answer forum

1.4 Existing Solution

There are many web application form where we can review or used code according to our needs. For Example, (stack overflow, html-css, guru 99) but we are not sure that the code is correct or not and sometimes nobody replies our quires properly regarding our problem. Therefore, we introduce **Wayforward**. We trained users to tell other users what they're doing wrong.

1.5 Proposed Solution:

When a fresh graduate student applies for job, always each and every company wants experiences person for the job. Nobody give a chance for fresh developer or student because, large companies are constantly working on numerous projects and captured the whole market. There're big teams of people, expert developers and few leaders responsible for making each project a successful endeavor. However, time marches on, technology evolves and everyone needs a technical solution for their business but no one wants to trust on fresh developer. There is a need of a platform for developers to share their skills, ideas, enhance skills and most important to earn something. So that the new/fresher developer can learn, work and gain Industry experience and get easily adjust in market.

Chapter 2

System Requirement Specification

2.1 Introduction

2.1.1 Purpose

This document provides requirement Specification for WayForward Web Application. Our Purpose is to support new developer and those who wants to learn industry experiences. System will available for the code sharing, live editing, question and answers, *E-commerce* anywhere and anytime.

2.1.2 Scope

The **Wayforward** is a web application includes all accessibilities to their users. This system is design in such a way that user can accessed from anywhere. For this user needs a web browser and a strong internet connection for used this system? All the queries and records will be managed to database through a web browser. User of this system are employees such as professors etc. The authorities in charge of managing approvals, rejections of codes by the admin.

2.1.3 Reference

- 1- Stack Overflow
- 2- Github
- 3- Upwork
- 4- Codeguru
- 5- <https://html-css-js.com/>
- 6- <https://www.paystubcreator.net/>

2.2 Overall Description

2.2.1 Product Perspective

WayForward: write the script of the future is an interactive web application. Our source of inspiration are those customers, software developers and their organization using the system. So, our system provides the solutions to the customers regarding coding. Some of the new codes have been generated by different customers and they allow others to see those codes. It will work dynamically as customers will also sell their codes.

2.2.1.1 System Interfaces

This should list each system interface and identify the functionality of the software to accomplish the system requirement and the interface description to match the system.

2.2.1.2 User Interfaces

- A. This product would contain numerous user interfaces ranging of that from web. The list below shows the user interfaces that are comprised in Webpages respectively. There are numerous features that we are trying to embed in this particular software so quite a few screens are used to show clarity amongst different interfaces.

Screens for Website

1. Login
2. Create Account
3. Forget ID or Password
4. Search Bar
5. Customer Profile View
6. Customer Details
7. Contact Customer
8. Report Customer
9. Login Activity
10. Related Problems
11. Languages
12. Add, Delete, Edit Codes
13. View Codes
14. Search Codes
15. Sell and Buy Codes
16. Payments
17. Tutorials
18. Comments
19. Projects
20. Editor/Compiler/Debugger
21. Search Methods

- B. Correct username and password must be entering into the system to make sure that the correct user is using the system. If for some odd reason the correct values are not entered, then the system would quickly prompt the user to enter correct data.

2.2.1.3 Hardware Interfaces

To run this system smoothly following hardware tools must be used:

Processor	Core i3/i5 dual core
Hard Disk	250 GB
RAM	2 GB
LAN Card:	
Keyboard	108 keys
Monitor	VGA

2.2.1.3.1 Computer

A PC with either Windows, Mac installed. A web browser is must installed to view this application. Recommended browsers are Google Chrome, Mozilla Firefox, Opera and Safari.

2.2.1.3.2 Internet Access Device

The internet connection should be stable. IP/domain of the system must be allowed in firewall rules if applicable.

2.2.1.4 Software Interfaces

Operating System	Windows, Mac
User/Web Interface	HTML5/CSS3/JavaScript/jQuery/React
Programming Language	PHP/Laravel
Database	MySQL Database
Browser	Google Chrome, Mozilla Firefox, Safari, Opera

2.2.1.5 Communications Interfaces

This is a web-based application so there is a communication way in the form of comments also there is a contact option through which different customers can contact to each other directly.

2.2.1.6 Memory

There are no constraints on either primary or secondary memory.

2.2.1.7 Operations

These operations are done by admin and user.

1. Admin add codes.
2. Admin sell codes.
3. Admin add users.
4. Admin manage all activities in the system.
5. User add codes.
6. User sell codes.
7. Database in SQL Format.
8. User Manual of System.

2.2.1.8 Site Adaptations Requirements

No special requirements are required for this site adaptation a user have any smartphone, laptop or desktop.

2.2.2 Product Functions

The users of the system are administrator, customer, and developer. The first one is admin and the second one is either developer or customer. They can identify the login activity of the system.

2.2.2.1 Administrator

Admin approaches a wide range of highlights of the site. He can add or erase the record. He has all admittance to the site. He can make changes in the information base. He approaches keep up the site. He has the position to make other individual as administrator.

2.2.2.2 User

User can deal with their record. User can add profile. They can change their data. User can add, delete, and update codes.

2.2.3 User Characteristics

There are three sorts of individuals that are cooperating with the framework: customer, administrator and organization. Every one of these users has distinctive utilization of framework so every one of them has their own necessities.

2.2.3.1 General Users

This user will be an ordinary user who will have the option to utilize the framework as per their own necessities. This user will have the option to perform register, login, search, add, erase, alter, see and different capacities as indicated by things.

2.2.3.2 Software Engineers

This user will have the option to manage the technical end like bug fixes, framework and implementation related issues.

2.2.3.3 Administrator

There is a technical person who shall have proper knowledge of the system and provides training and support to users.

2.2.4 Constraints

The main constraint for this system is internet connection. As this system can perform functions only if user login. So, the internet connection is important.

Some of The Constraints Are As Follows:

2.2.4.1 Cost

Cost of the project is defined by the owner; the project should be completed in 20 thousand to 40 thousands.

2.2.4.2 Schedule/time

Time is fixed by the owner and the owner gives the time period of 8 months to complete the project.

2.2.4.3 System Software Constraints

Minimum eligibility for this system is windows 7. It can run in higher versions smoothly.

2.2.5 Assumptions and Dependencies

This product is used only if there is an internet connection. Some of the other are as follows:

a) Budget

The developer's team estimates all cost of the product is around 50 thousand.

b) Schedule of reporting

Appointments between the project team and owner regarding project and they assume that it will complete in 8 months.

c) Resource Availability

System manager should provide resources which are suitable for the project team considering the complexity of project.

d) Environment/Circumstances

Project team should have developed software which needs to appropriate for the environment. On which conditions project will be utilized, they need to think about every one of these circumstances and need to satisfy them.

e) Technical (Technology)

Hardware and web application both framework would have that adaptability to entertain themselves with terms of advancements.

f) Dependencies

Project dependencies should be followed, similar to extend finish-start dependencies. All the exercises of our project from breaking down to deploying should be followed step by step.

g) Security

All system activities are under inspection through databases they maintain the log of activities.

2.2.6 Apportioning of requirements

If there are any updates in the software, we may need a few additional requirements to cope with the problem.

2.3 Specific Requirements

2.3.1 Functional Requirements

Functional requirement is used to define a function of a system and its components. Functional requirements of RMS web application may be its technical details, data manipulation and processing and other specific functionality that define “what is system going to do”.

There are two main modules in our project:

- 1. Admin**
- 2. User**
 - 1. Login**
 - a. Login**
 - b. Forget id and password**

ID:	FR_01		
Name:	Login		
Description	Input	Output	Basic work flow
It will let the user to login in the system and shows the screens according to their roles.	Id and password	User logged in to the system.	Authenticated user will fill the login form. Submit the login form. System verifies the id and password. Dashboard of the user appears on the screen where user can perform desired tasks.

Table 1 Functional Requirement Login

ID:	FR_02		
Name:	Forget ID OR password		
Description	Input	Output	Basic work flow
User forgets his password or id then he will recover it.	User ID	Password or ID access able	User logged in to the system. Dashboard appeared on the screen. User click on User Tab. User list will be appeared on the screen. Select the user for reset password. Click on Edit. Reset the Password or Id. Send Mail to User.

Table 2 Functional Requirement Forget ID or Password

2. Administrative management system

- a. Register new user
- b. Approve user registration
- c. Block users
- d. Assign specific roles
- e. Approve user questions and answers
- f. Remove user queries
- g. Approve developers' scripts

ID:	FR_03		
Name:	Register new user		
Description	Input	Output	Basic work flow
User will be added in the system list and will access their roles.	Click on signup button on the website.	User has their id password and role.	<p>Dashboard will be open.</p> <p>Click on signup button on website.</p> <p>User type list will be appeared on the screen.</p> <p>Form will be open for register user.</p> <p>Fill the information form of the user.</p> <p>Assign him their roles.</p> <p>Click on save button.</p>

Table 3 Functional Requirement Register New User

ID:	FR_04		
Name:	Approve user registration		
Description	Input	Output	Basic work flow
It will approve the registration request from the user.	Click on approve button.	User approved	<p>Click on the notification bar visible for admin.</p> <p>Click on the registration request.</p> <p>Verify the user information.</p> <p>Click on the approve button.</p> <p>Approve confirmation page will be open.</p> <p>Click on OK to approve user.</p>

Table 4 Functional Requirement Approve User Registration

ID:	FR_05		
Name:	Block user		
Description	Input	Output	Basic work flow
The user will be block from the record and they have no more authentications for access any role.	Click on delete button.	User blocked.	Admin click on User Tab. User list will be appeared on the screen. Enter User ID in search tab. Click on user ID. Click on Delete button. Deletion Confirm page will be open. Click on OK to Delete user.

Table 5 Functional Requirement Block User

ID:	FR_06		
Name:	Assign specific role		
Description	Input	Output	Basic work flow
It will assign the specific role to a specific user.	Click on assign button.	Role assigned to the user.	Admin click on the user tab. User list will be appeared on the screen. Enter user ID in the search tab. Click on user ID. Select the specific role to allocate to that user. Click on assign button. Assigning role confirmation page will be open. Click on OK to assign role.

Table 6 Functional Requirement Assign specific role

ID:	FR_07		
Name:	Approve user questions and answers		
Description	Input	Output	Basic work flow
It will approve the queries and solution of that query from the users.	Click on the approve button.	Questions and answers approved.	Click on the query tab. Click on the notification bar. Select the query ask and solution provided by user. Click on approve button. Approve confirmation page will be open. Click on OK to approve query and solution.

Table 7 Functional Requirement Approve user questions and answers

ID:	FR_08		
Name:	Remove user queries		
Description	Input	Output	Basic work flow
It will remove the unnecessary queries asked by user.	Click on the remove button.	Queries are removed.	Click on the query tab. Click on the notification bar. Select the query asked by the user. Click on unnecessary query. Click on remove button. Removed confirmation page will be open. Click on OK to remove query

Table 8 Functional Requirement Remove user queries

ID:	FR_09		
Name:	Approve developer's scripts		
Description	Input	Output	Basic work flow
It will approve the user/developer script/code.	Admin click on approve button.	Developer's script approved.	Admin shall click on the main code tab. Click on the notification bar. Select the script of developer. Click on approve button. Approved confirmation page will be open. Click on OK to approve script.

Table 9 Functional Requirement Approve developer's scripts

3. Live Editor Management system

- a. Write code on live editor
- b. Save code on local machine
- c. Allow registered user to save code in repository

ID:	FR_10		
Name:	Write code on live editor		
Description	Input	Output	Basic work flow
It will write the code of specific problem on the live editor/compiler.	Click on the compiler tab. Write code on the live editor.	Code successfully run.	Click on the compiler tab on the website. Select the languages available. Write the code on the compiler. Outputs has been shown at that moment.

Table 10 Functional Requirement Write code on live editor

ID:	FR_11		
Name:	Save code on local machine		
Description	Input	Output	Basic work flow
It will save the code on compiler temporary.	Click on the save button.	Code saved on local machine.	Click on the save button. Click on the local machine tab.

Table 11 Functional Requirement Save code on local machine

ID:	FR_12		
Name:	Allow registered user to save code in repository		
Description	Input	Output	Basic work flow
It will save the code to database from compiler.	Click on the save as button. Click on the database.	Code saved on the database.	Click on the save as button. Options will be appear on the screen. Select the desire option. Click on the database. Click on the confirm button.

Table 12 Functional Requirement Allow registered user to save code in repository

4. Repository management system

- a. Edit user code
- b. View user code Detail
- c. Add code to database
- d. View code on database
- e. Delete code

ID:	FR_13		
Name:	Edit user code		
Description	Input	Output	Basic work flow
It will update the user code.	Click on user record button. Click on user code tab. Click on edit user code button.	User's code updated.	Click on edit user code button. Edit user code page will be open. Edit and update the code of the user.

Table 13 Functional Requirement Edit user code

ID:	FR_14		
Name:	View user code detail		
Description	Input	Output	Basic work flow
System will view individual detail of the user's code from the list of all registered users.	Click on user code tab. Click on detail user code button.	View individual user code detail list	Click on user code Tab. User code list will be appeared on the screen. Click on Detail user code button. Detail page of the user will be appeared on the screen.

Table 14 Functional Requirement View user code detail

ID:	FR_15		
Name:	Add code to database		
Description	Input	Output	Basic work flow
The code with description will be added in the database.	Code of problem.	Code is on database.	Admin click on Main code Tab. Current code list will be open. Click on Add code button. Form will be open for add new code or select from available code from website which is temporary before. Click on the save button.

Table 15 Functional Requirement Add code to database

ID:	FR_16		
Name:	View code on database		
Description	Input	Output	Basic work flow
It will view the individual detail of the code from list of all code.	Click on main code Tab. Click on detail button.	View the code.	Click on the main code tab. Current code list will be open. All the codes will appear on screen. Click on detail button. Detail page of the code will be appeared on the screen.

Table 16 Functional Requirement View code on database

ID:	FR_17		
Name:	Delete code on database		
Description	Input	Output	Basic work flow
It will delete the codes from the record/database.	Click on main code tab. Click on delete button of selected code.	Code deleted	Click on Main code Tab. Current code list will be open. All the codes will appear on screen. Select the certain code which user/admin want to delete. Click on Delete button of the selected item.

Table 17 Functional Requirement Delete code on database

5. Sales management system

- a. Payment gateway intergradation
- b. Sale and purchase management
- c. Multiple currency option
- d. Allow developer to sale code
- e. Allow users to purchase code
- f. Payment report generation

ID:	FR_18		
Name:	Payment gateway integration		
Description	Input	Output	Basic work flow
It will allow user to purchase code with multiple methods available in country.	Multiple methods.	User select his method.	Select the code. Click on the purchase code. System ask for payment method options. User selects method. System ask for account option. User enter his available account. User click on confirm button.

Table 18 Functional Requirement Payment gateway integration

ID:	FR_19		
Name:	Sale and purchase management		
Description	Input	Output	Basic work flow
It will help admin to manage sales and purchase.	Admin has detail of users ID.	Sales and purchase report.	Details of the sales and purchase available to the admin. Detail of the developer with payment available to the admin. Detail of user with code available to the admin. Admin confirms that the user has code and developer has payment.

Table 19 Functional Requirement Sale and purchase management

ID:	FR_20		
Name:	Multiple currency option		
Description	Input	Output	Basic work flow
It will allow user to purchase code with multiple currency.	Currency	User select his currency.	Select the code. Click on the purchase code. System ask for currency options. User enter his currency option. System ask for card option. User enter his available card. User click on confirm button.

Table 20 Functional Requirement Multiple currency option

ID:	FR_21		
Name:	Allow developer to sale code		
Description	Input	Output	Basic work flow
It will allow developer to sale his code to others.	Developer make a sales.	Developer code is on sale.	<p>Developer shall click on the sales tab.</p> <p>Developer shall click on the upload code.</p> <p>Developer add code with problem.</p> <p>Developer shall click on the add price button.</p> <p>Add the price of code.</p> <p>Click on public code with price for sale.</p>

Table 21 Functional Requirement Allow developer to sale code

ID:	FR_22		
Name:	Allow users to purchase code		
Description	Input	Output	Basic work flow
It will allow user to purchase a code of specific problem from the developer.	User click on purchase code button.	User have his required code.	<p>User shall click on sales tab</p> <p>User can search his problem on search bar.</p> <p>User see if the solution is available.</p> <p>User click on purchase code.</p> <p>User make a payment method.</p> <p>User give the payment.</p>

Table 22 Functional Requirement Allow users to purchase code

ID:	FR_23		
Name:	Payment report generation		
Description	Input	Output	Basic work flow
It will allow system to generate a payment report.	Payment details.	Payment report	<p>Admin shall click on the sales tab.</p> <p>Admin can view the payments.</p> <p>Admin shall select the payment.</p> <p>Click on the generate report button.</p> <p>Report of the payment has been opened.</p>

Table 23 Functional Requirement Payment report generation

6. Feedback management system

- a. Review on codes
- b. Rating

ID:	FR_24		
Name:	Review on codes		
Description	Input	Output	Basic work flow
It will help the user or admin to take feedback on codes.	Comments on the certain code.	Comment on the code.	<p>Click on the main code tab.</p> <p>Current code list will be open.</p> <p>Click on the code.</p> <p>Click on the comment button.</p> <p>Write the feedback on selected code.</p> <p>Click on the save button.</p>

Table 24 Functional Requirement Review on codes

ID:	FR_25		
Name:	Code rating		
Description	Input	Output	Basic work flow
It will help the user or admin to see acceptance of codes.	Rate the specific code.	Code has been rated.	Click on the main code tab. Current code list will be open. Click on the code. Click on the stars for rating the code. Click on the save button.

Table 25 Functional Requirement Code rating

7. Problem solving management system

- a. Ask query
- b. Solution of query

ID:	FR_26		
Name:	Ask query		
Description	Input	Output	Basic work flow
It will help the users of website to ask questions from other users or from admin.	Specific problem.	Question asked from other users.	Click on the query tab. Click on the ask question button. Write the problem or query. Click on done button.

Table 26 Functional Requirement Ask query

ID:	FR_27		
Name:	Solution of query		
Description	Input	Output	Basic work flow
It will provide the solution of specific problem.	Solution of problem.	Solution is available of specific problem.	Click on the query tab. Select the specific problem which solution is available. Click on the answer button. Place the solution of query. Click on the save button.

Table 27 Functional Requirement Solution of query

8. Code sharing management system

- a. Admin upload code
- b. User upload code

ID:	FR_28		
Name:	Admin upload code		
Description	Input	Output	Basic work flow
Admin can upload the code on the website and description of the code will be added along code.	Code of certain requirements.	Code on website.	Admin shall click on the code tab. Click on the add code button. System shall allocate space for code. Admin shall place the code on allocated space. Admin click on upload button.

Table 28 Functional Requirement Admin upload code

ID:	FR_29		
Name:	User upload code		
Description	Input	Output	Basic work flow
User can upload the right code on the website and description of the code will be added.	Code of certain requirements .	Code on website.	User shall click on the code tab. User shall click on the add code button. System shall allocate space for code. User shall place the code on allocated space. User click on upload button.

Table 29 Functional Requirement User upload code

9. User management system

- a. Upload script
- b. Update script
- c. Delete script
- d. Allow user to ask queries
- e. Allow user to participate in voting
- f. Switch roles
- g. Allow users to rate scripts

ID:	FR_30		
Name:	Upload script		
Description	Input	Output	Basic work flow
User can upload the script on website with their description.	Script of certain problem.	Script on website.	<p>User shall click on the main code tab.</p> <p>User shall click on the add code button.</p> <p>System shall allocate space for script.</p> <p>User shall place the script on allocated space.</p> <p>User click on upload button</p>

Table 30 Functional Requirement Upload script

ID:	FR_31		
Name:	Update script		
Description	Input	Output	Basic work flow
It will updated the user script.	Click on user record button. Click on the user code tab. Click on edit user code button.	Developer's script updated.	Click on edit user code button. Edit developer's script. Click on the update button. Update confirmation button will be open. Click on OK for update script.

Table 31 Functional Requirement Update script

ID:	FR_32		
Name:	Delete script		
Description	Input	Output	Basic work flow
It will delete the script of developer.	Click on the delete button of selected script.	Script deleted.	Click on main code tab. Current script list will be open. Select the script for deletion. Click on the delete button of the selected script. Delete confirmation page will open. Click on OK button to delete.

Table 32 Functional Requirement Delete script

ID:	FR_33		
Name:	Allow user to ask questions		
Description	Input	Output	Basic work flow
It will help user to ask question for solution.	Problem	Question asked from other users or admin.	Click on the query tab. Click on the ask question button. Write the problem or question. Click on done button.

Table 33 Functional Requirement Allow user to ask questions

ID:	FR_34		
Name:	Allow user to participate in voting		
Description	Input	Output	Basic work flow
It will help user to vote for a code for code availability on system.	Click on voting.	Script has been voted.	Click on the main code tab. Current code list will be open. Click on the code. Click on the voting code. Vote "YES" or "NO". Click on the save button.

Table 34 Functional Requirement Allow user to participate in voting

ID:	FR_35		
Name:	Allow user to Switch roles		
Description	Input	Output	Basic work flow
It will allow user to switch their role from user to developer or from developer to user.	Click on switch button.	Switch role successfully.	<p>User shall click on my details tab.</p> <p>User shall click on role button.</p> <p>User shall click on switch role button.</p> <p>Select the role.</p> <p>Role selection confirmation page will be open.</p> <p>Click on OK button to switch role.</p>

Table 35 Functional Requirement Allow user to Switch roles

ID:	FR_36		
Name:	Allow users to rate scripts		
Description	Input	Output	Basic work flow
It will help user to rate a code for better.	Click on rating.	Script has been rated.	<p>Click on the main code tab.</p> <p>Current script list will be open.</p> <p>Click on the script.</p> <p>Click on the stars for rating the script.</p> <p>Click on the save button.</p>

Table 36 Functional Requirement Allow users to rate scripts

2.3.2 Non Functional Requirements

The non-functional requirement explains a performance characteristic of the system.

2.3.2.1 System Performance

The system performance is fully satisfied and can run smoothly on different devices. If user wants to perform more than one functions at same time, then it will not affect performance and speed of system.

2.3.2.2 Availability

It refers to a property of software that it is there and ready to carry out its task when you need it to be. If server is down in any case, then it will give access to you in few hours.

2.3.2.3 Integrity

It is the quality of the software's source code that no unauthorized changes have been made to the delivered software.

2.3.2.4 Scalability

It is an attribute of a tool or a system to increase its capacity and functionalities based on its users demand.

2.3.2.5 Reliability

The system is error free as it operates normally in different conditions in specified environments. The site will guarantee the security of client's status and guarantee full control over execution, so asset designation is preposterous without administrator authority.

2.3.2.6 Maintainability

The system can be maintainable after the deployment. It refers to the ease with which we can repair, improve and understand software code.

2.3.2.7 Security

The system is fully secure as it needs user confirmation before doing any activity. If a user close browser unintentionally then the system will automatically close users account and ask user to enter the credentials again.

2.3.2.8 Usability

Usage of the framework is high since it is anything but difficult to utilize. Names of decisions are very clear, as indicated by their usefulness. Its interface isn't troublesome as client can see everything obviously and utilize its usefulness without any confusion.

Chapter 3

Use Case Analysis

Administrator use case diagram

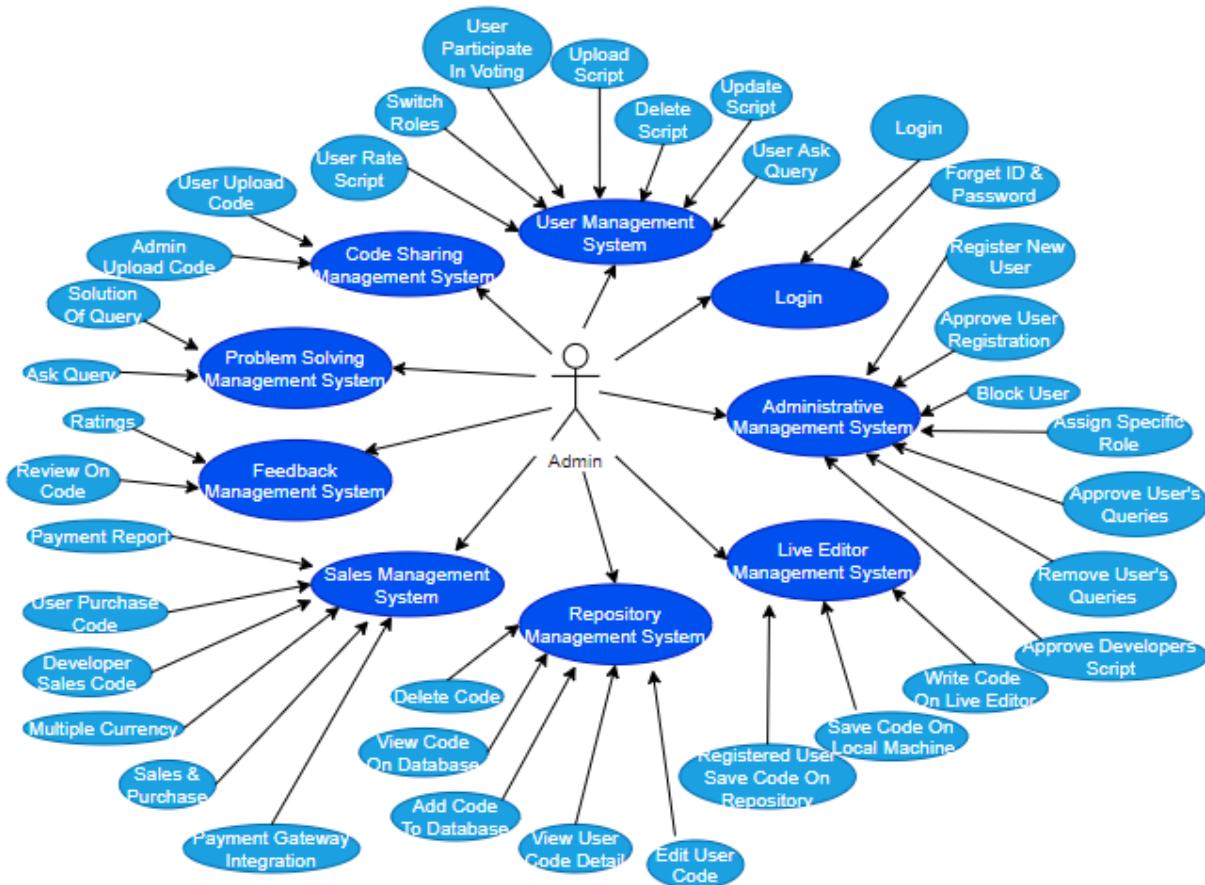


Figure 1 Administrator use case diagram

3.1 Administrator Use Case Diagrams & Specification

3.1.1 UC Login

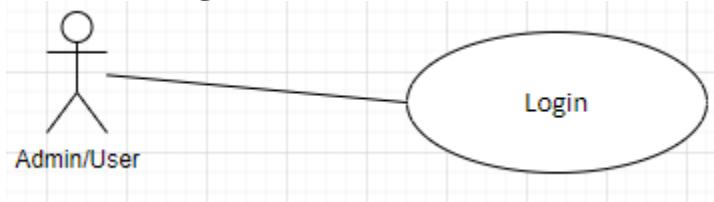


Figure 2 Use Case Login

Section	Detail
ID	UC-01
FR ID	FR_01
Name	Login
Description	It will let the user to login in the system and shows the screens according to their roles.
Actor	Primary: Admin, User
Precondition	User must have their id and password.
Post Conditions (Success Guarantee)	User logged in to the system.
Basic Flow (Main Success Scenario)	<ol style="list-style-type: none"> Enter the portal address. Login page will be open. Authenticated user will fill the login form. Submit the login form. System verifies the id and password. Dashboard of the user appears on the screen where user can perform desired tasks.
Alternative Flow	<ol style="list-style-type: none"> System fails to access portal address <ol style="list-style-type: none"> To support the system recovery, restart the system and request for the initial state. Login page is not available <ol style="list-style-type: none"> Check the internet connection and refresh the portal.

Table 37 Use Case Specification Login

3.1.2 UC Forget ID or Password



Figure 3 Use Case Forget ID or Password

Section	Detail
ID	UC-02
FR ID	FR_01, FR_02
Name	Forget ID or Password
Description	User forgets his password or id then he will recover it.
Actor	Primary: Admin, User
Precondition	User must have their ID.
Post Conditions (Success Guarantee)	The system successfully changed the login password.
Basic Flow (Main Success Scenario)	<ol style="list-style-type: none"> 1. User logged in to the system. 2. Dashboard appeared on the screen. 3. User click on User Tab. 4. User list will be appeared on the screen. 5. Select the user for reset password. 6. Click on Edit. 7. Reset the Password or Id. 8. Send Mail to User.
Alternative Flow	<ol style="list-style-type: none"> 1. User is not registered to the system yet <ol style="list-style-type: none"> a. Contact to the admin via email for username and password. 2. System show the error message that username or password is incorrect <ol style="list-style-type: none"> a. Re-enter the username and password 3. Login request is cancelled due to the poor internet connection resend the request

Table 38 Use Case Specification Forget ID or Password

3.1.3 UC Register New User



Figure 4 Use Case Register new user

Section	Detail
ID	UC-03
FR ID	FR_03
Name	Register new user
Description	User will be added in the system list and will access their roles.
Actor	Primary: Admin
Precondition	User who want to be a part of website has accessibility on signup interface.
Post Conditions (Success Guarantee)	New user successfully registered to the system.
Basic Flow (Main Success Scenario)	<ol style="list-style-type: none"> 1. Dashboard will be open. 2. Click on signup button on website. 3. User type list will be appeared on the screen. 4. Form will be open for register user. 5. Fill the information form of the user. 6. Assign him their roles. 7. Click on save button.
Alternative Flow	<ol style="list-style-type: none"> 1. User Already Exist. <ol style="list-style-type: none"> a. Show Error message b. Update the user. 2. No option available on Administrator dashboard <ol style="list-style-type: none"> a. Admin don't have permission to manage user. b. System will show the error message to admin. c. Admin will contact the developer for permission. d. Developer will give permission to admin. 3. If given information is not valid than <ol style="list-style-type: none"> a. System will show the pop up to user with error detail b. User will check the problem with given data c. System will also show the valid information format and pattern.

Table 39 Use Case Specification Register new user

3.1.4 UC Approve User Registration



Figure 5 Use Case Approve User Registration

Section	Detail
ID	UC-04
FR ID	FR_03, FR_04
Name	Approve User Registration
Description	It will approve the registration request from the user.
Actor	Primary: Admin
Precondition	Registration request of the user generated.
Post Conditions (Success Guarantee)	User has access on the website and has his role.
Basic Flow (Main Success Scenario)	<ol style="list-style-type: none"> Click on the notification bar visible for admin. Click on the registration request. Verify the user information. Click on the approve button. Approve confirmation page will be open. Click on OK to approve user.
Alternative Flow	<ol style="list-style-type: none"> User Already Exist. <ol style="list-style-type: none"> Show Error message Update the user. If given information is not valid than <ol style="list-style-type: none"> System will show the pop up to Admin with error detail Admin will check the problem with given data System will also show the valid information format and pattern.

Table 40 Use Case Specification Approve User Registration

3.1.5 UC Block User

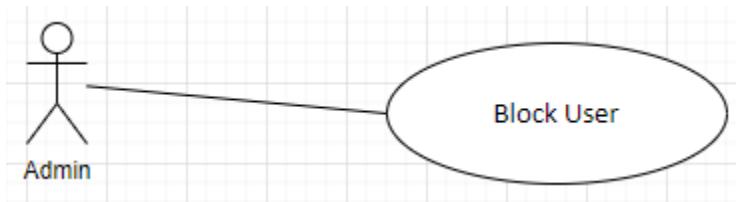


Figure 6 Use Case Block User

Section	Detail
ID	UC-05
FR ID	FR_05
Name	Block User
Description	The user will be block from the record and they have no more authentications for access any role.
Actor	Primary: Admin
Precondition	Admin must have user ID.
Post Conditions (Success Guarantee)	User has no more access on his role.
Basic Flow (Main Success Scenario)	<ol style="list-style-type: none"> 1. Admin click on User Tab. 2. User list will be appeared on the screen. 3. Enter User ID in search tab. 4. Click on user ID. 5. Click on Delete button. 6. Deletion Confirm page will be open. 7. 1.7 Click on OK to delete user.
Alternative Flow	<ol style="list-style-type: none"> 1. If given information is not valid than <ol style="list-style-type: none"> a. System will show the pop up to admin with error detail. b. Admin will check the problem with given data c. System will also show the valid information format and pattern to admin d. Admin will flow the pattern and click again on the Block user button. 2. Delete Cancelled <ol style="list-style-type: none"> a. The delete is cancelled, and the Basic Flow is re-started at the beginning. 3. Deletion Error <ol style="list-style-type: none"> a. System will show the error message. b. Back to the user list.

Table 41 Use Case Specification Block User

3.1.6 UC Assign Specific Role

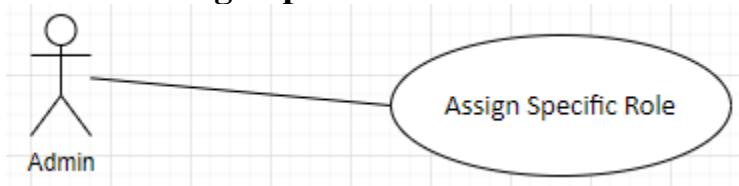


Figure 7 Use Case Assign Specific Role

Section	Detail
ID	UC-06
FR ID	FR_03, FR_06
Name	Assign Specific Role
Description	It will assign the specific role to a specific user.
Actor	Primary: Admin
Precondition	Admin must have user ID.
Post Conditions (Success Guarantee)	User have specific role.
Basic Flow (Main Success Scenario)	<ol style="list-style-type: none"> 1. Admin click on the user tab. 2. User list will be appeared on the screen. 3. Enter user ID in the search tab. 4. Click on user ID. 5. Select the specific role to allocate to that user. 6. Click on assign button. 7. Assigning role confirmation page will be open. 8. Click on OK to assign role.
Alternative Flow	<ol style="list-style-type: none"> 1. If all roles are assigned <ol style="list-style-type: none"> a. System will show the pop up to admin will role details b. Admin will check the problem with the roles c. Admin increase the role for the user. d. Admin will assign the role.

Table 42 Use Case Specification Assign Specific Role

3.1.7 UC Approve User's Queries



Figure 8 Use Case Approve User's Queries

Section	Detail
ID	UC-07
FR ID	FR_07, FR_26
Name	Approve User's Queries
Description	It will approve the queries and solution of that query from the users.
Actor	Primary: Admin
Precondition	Question asked by user. Solution provided by user.
Post Conditions (Success Guarantee)	Queries and solution available for other users.
Basic Flow (Main Success Scenario)	<ol style="list-style-type: none"> Click on the query tab. Click on the notification bar. Select the query ask and solution provided by user. Click on approve button. Approve confirmation page will be open. Click on OK to approve query and solution.
Alternative Flow	<ol style="list-style-type: none"> If the query is not valid <ol style="list-style-type: none"> Click on the remove query button. If the query isn't in website language <ol style="list-style-type: none"> Click on the edit query button Change the query in appropriate words and language. Then click on the approve button.

Table 43 Use Case Specification Approve User's Queries

3.1.8 UC Remove User's Queries

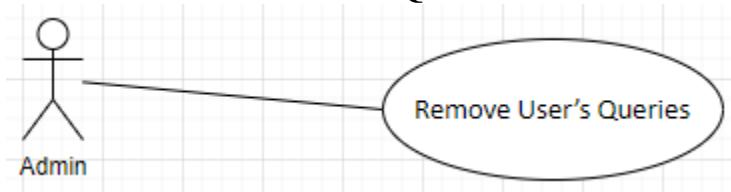


Figure 9 Use Case Remove user's queries

Section	Detail
ID	UC-08
FR ID	FR_08, FR_26
Name	Remove user's queries
Description	It will remove the unnecessary queries asked by user.
Actor	Primary: Admin
Precondition	Queries asked by user.
Post Conditions (Success Guarantee)	User has notification that his/her query has been removed.
Basic Flow (Main Success Scenario)	<ol style="list-style-type: none"> 1. Click on the query tab. 2. Click on the notification bar. 3. Select the query asked by the user. 4. Click on unnecessary query. 5. Click on remove button. 6. Removed confirmation page will be open. 7. Click on OK to remove query
Alternative Flow	<ol style="list-style-type: none"> 1. Deletion Error <ol style="list-style-type: none"> a. System will show the error message. b. Back to the queries list. 2. If the query is not valid <ol style="list-style-type: none"> a. Click on the remove query button.

Table 44 Use Case Specification Remove user's queries

3.1.9 UC Approve Developer' Script



Figure 10 Use Case Approve Developer' Script

Section	Detail
ID	UC-09
FR ID	FR_09, FR_12, FR_29, FR_30
Name	Approve Developer' Script
Description	It will approve the user/developer script/code.
Actor	Primary: Admin
Precondition	Script added by the developer.
Post Conditions (Success Guarantee)	Developer's script is on website.
Basic Flow (Main Success Scenario)	<ol style="list-style-type: none"> 1. Admin shall click on the main code tab. 2. Click on the notification bar. 3. Select the script of developer. 4. Click on approve button. 5. Approved confirmation page will be open. 6. Click on OK to approve script.
Alternative Flow	<ol style="list-style-type: none"> 1. If the script isn't correct <ol style="list-style-type: none"> a. Send a mail to developer to change his code or correct it. 2. If script already available on website <ol style="list-style-type: none"> a. Do not approve Developer' script.

Table 45 Use Case Specification Approve Developer' Script

3.1.10 UC Write Code on Live Editor

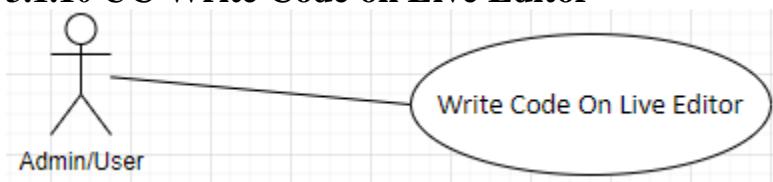


Figure 11 Use Case Write Code On Live Editor

Section	Detail
ID	UC-10
FR ID	FR_10
Name	Write Code On Live Editor
Description	It will write the code of specific problem on the live editor/compiler.
Actor	Primary: Admin, User
Precondition	Compiler available on the website.
Post Conditions (Success Guarantee)	Code with the description and output has been wrote on compiler.
Basic Flow (Main Success Scenario)	<ol style="list-style-type: none"> Click on the compiler tab on the website. Select the languages available. Write the code on the compiler. Outputs has been shown at that moment.
Alternative Flow	<ol style="list-style-type: none"> System fails to access portal address <ol style="list-style-type: none"> To support the system recovery, restart the system and request for the initial state. Live editor page is not available <ol style="list-style-type: none"> Check the internet connection and refresh the portal.

Table 46 Use Case Specification Write Code On Live Editor

3.1.11 UC Save Code on Local Machine

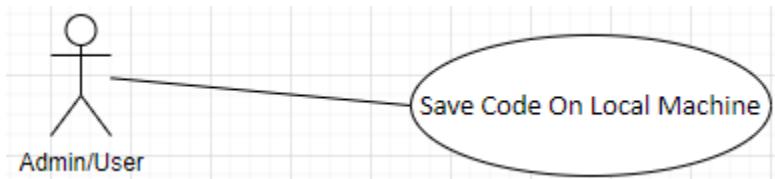


Figure 12 Use Case Save Code On Local Machine

Section	Detail
ID	UC-11
FR ID	FR_10, FR_11
Name	Save Code On Local Machine
Description	It will save the code on compiler temporary.
Actor	Primary: Admin, User
Precondition	Code has been written on live editor.
Post Conditions (Success Guarantee)	Code successfully saved on local machine/compiler.
Basic Flow (Main Success Scenario)	<ol style="list-style-type: none"> Click on the save button. Click on the local machine tab.
Alternative Flow	<ol style="list-style-type: none"> Live editor page is not working properly <ol style="list-style-type: none"> Check the internet connection and refresh the portal.

Table 47 Use Case Specification Save Code On Local Machine

3.1.12 UC Allow Registered User to Save Code In Repository

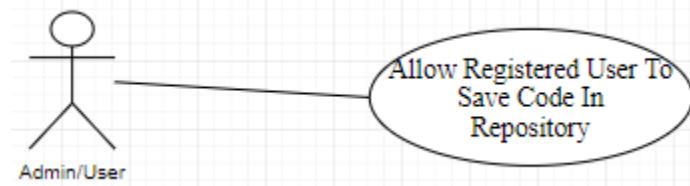


Figure 13 Use Case Allow Registered User To Save Code In Repository

Section	Detail
ID	UC-12
FR ID	FR_10, FR_11, FR_12
Name	Allow Registered User To Save Code In Repository
Description	It will save the code to database from compiler.
Actor	Primary: Admin, User
Precondition	Code has been written on the live editor.
Post Conditions (Success Guarantee)	Code successfully saved on to database from local machine.
Basic Flow (Main Success Scenario)	<ol style="list-style-type: none"> 1. Click on the save as button. 2. Options will be appear on the screen. 3. Select the desire option. 4. Click on the database. 5. Click on the confirm button.
Alternative Flow	<ol style="list-style-type: none"> 1. If the code isn't correct <ol style="list-style-type: none"> a. A pop up to user to change his code or correct it. 2. If code is already available on system <ol style="list-style-type: none"> a. A pop up to user that code is already available on database.

Table 48 Use Case Specification Allow Registered User To Save Code In Repository

3.1.13 UC Edit User Code

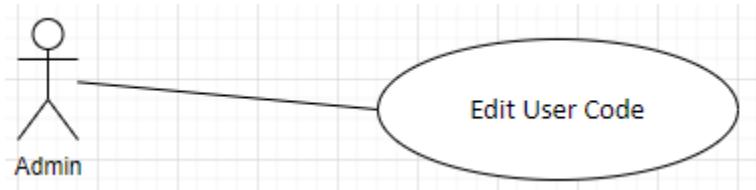


Figure 14 Use Case Edit User Code

Section	Detail
ID	UC-13
FR ID	FR_12, FR_15, FR_13, FR_29, FR_30, FR_31
Name	Edit User Code
Description	It will update the user code.
Actor	Primary: Admin
Precondition	User's code exists on website.
Post Conditions (Success Guarantee)	Updated code in on the website.
Basic Flow (Main Success Scenario)	<ol style="list-style-type: none"> Click on edit user code button. Edit user code page will be open. Edit and update the code of the user.
Alternative Flow	<ol style="list-style-type: none"> If the code isn't working <ol style="list-style-type: none"> Click on the edit user code button. Code details aren't added <ol style="list-style-type: none"> Compile the code and write description for it.

Table 49 Use Case Specification Edit User Code

3.1.14 UC View User Code Details

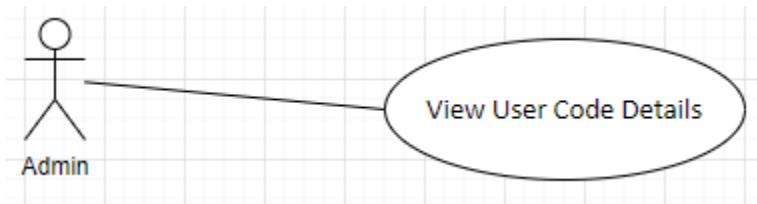


Figure 15 Use Case View User Code Details

Section	Detail
ID	UC-14
FR ID	FR_12, FR_15, FR_29, FR_30, FR_31
Name	View User Code Details
Description	System will view individual detail of the user's code from the list of all registered users.
Actor	Primary: Admin
Precondition	Registered user available on website and upload their codes.
Post Conditions (Success Guarantee)	User code details has been viewed by admin.
Basic Flow (Main Success Scenario)	<ol style="list-style-type: none"> Click on main code Tab. User code list will be appeared on the screen. Click on Detail user code button. Detail page of the user will be appeared on the screen.
Alternative Flow	<ol style="list-style-type: none"> No option available on Administrator dashboard <ol style="list-style-type: none"> Admin doesn't have permission to manage User code. System will show the error message to admin. Admin will contact the Developer for permission. Developer will give permission to admin. If given information is not valid than <ol style="list-style-type: none"> System will show the pop up to admin with error detail. Admin will check the problem with given data System will also show the valid information form at and pattern to admin Admin will flow the pattern and click again on the Detail user code button.

Table 50 Use Case Specification View User Code Details

3.1.15 UC Add Code to Database



Figure 16 Use Case Add Code To Database

Section	Detail
ID	UC-15
FR ID	FR_12, FR_15, FR_29, FR_30, FR_31
Name	Add Code To Database
Description	The code with description will be added in the database.
Actor	Primary: Admin
Precondition	Admin have code of problem along description.
Post Conditions (Success Guarantee)	Code will be added in the database.
Basic Flow (Main Success Scenario)	<ol style="list-style-type: none"> 1. Admin click on Main code Tab. 2. Current code list will be open. 3. Click on Add code button. 4. Form will be open for add new code or select from available code from website which is temporary before. 5. Click on the save button.
Alternative Flow	<ol style="list-style-type: none"> 1. Code Already Exist. <ol style="list-style-type: none"> a. Show Error message b. Update the code. 2. If the code isn't correct <ol style="list-style-type: none"> a. A pop up to user to change his code or correct it.

Table 51 Use Case Specification Add Code To Database

3.1.16 UC View Code on Database

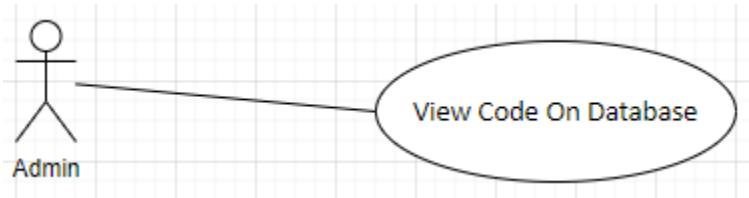


Figure 17 Use Case View Code On Database

Section	Detail
ID	UC-16
FR ID	FR_12, FR_29, FR_30, FR_31
Name	View Code On Database
Description	It will view the individual detail of the code from list of all code.
Actor	Primary: Admin
Precondition	Code exists on database.
Post Conditions (Success Guarantee)	Code has been viewed on repository.
Basic Flow (Main Success Scenario)	<ol style="list-style-type: none"> Click on the main code tab. Current code list will be open. All the codes will appear on screen. Click on view code button. Detail page of the code will be appeared on the screen.
Alternative Flow	<ol style="list-style-type: none"> If given information is not valid than <ol style="list-style-type: none"> System will show the pop up to admin with error detail. Admin will check the problem with given data System will also show the valid information format and pattern to admin Admin will flow the pattern and click again on the View code button. View code button is not visible <ol style="list-style-type: none"> Admin send request to the developer Developer check the problem and fix it Can't find the code <ol style="list-style-type: none"> Search for the user by its name System displays no result Admin send new record addition request to developer Developer will fix it and return Notification back to Admin

Table 52 Use Case Specification View Code On Database

3.1.17 UC Delete Code

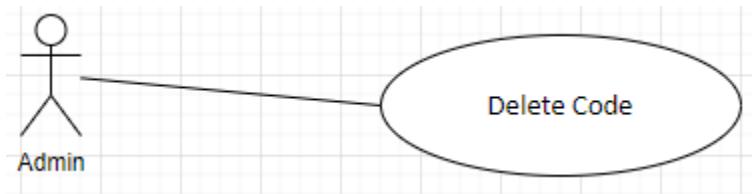


Figure 18 Use Case Delete Code

Section	Detail
ID	UC-17
FR ID	FR_12, FR_13, FR_29, FR_30, FR_31
Name	Delete Code
Description	It will delete the codes from the record/database.
Actor	Primary: Admin
Precondition	Code exists in database.
Post Conditions (Success Guarantee)	Code will be deleted from the database.
Basic Flow (Main Success Scenario)	<ol style="list-style-type: none"> Click on Main code Tab. Current code list will be open. All the codes will appear on screen. Select the certain code which user/admin want to delete. Click on Delete button of the selected item.
Alternative Flow	<ol style="list-style-type: none"> If given information is not valid than <ol style="list-style-type: none"> System will show the pop up to admin with error detail. Admin will check the problem with given data System will also show the valid information format and pattern to admin Admin will flow the pattern and click again on the Delete code button. Delete Cancelled <ol style="list-style-type: none"> The delete is cancelled, and the Basic Flow is re-started at the beginning. Deletion Error <ol style="list-style-type: none"> System will show the error message. Back to the code list.

Table 53 Use Case Specification Delete Code

3.1.18 UC Payment Gateway Integration

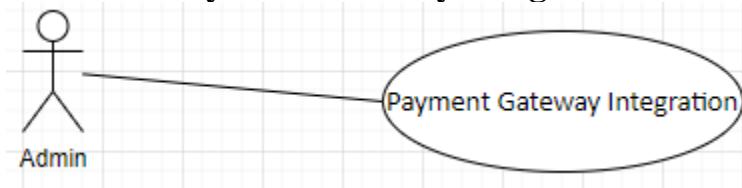


Figure 19 Use Case Payment Gateway Integration

Section	Detail
ID	UC-18
FR ID	FR_18, FR_19, FR_20, FR_21
Name	Payment Gateway Integration
Description	It will allow user to purchase code with multiple methods available in country.
Actor	Primary: Admin, User
Precondition	Payment option available.
Post Conditions (Success Guarantee)	Payment has been done with currency chosen by user.
Basic Flow (Main Success Scenario)	<ol style="list-style-type: none"> 1. Select the code. 2. Click on the purchase code. 3. System ask for payment method options. 4. User selects method. 5. System ask for account option. 6. User enter his available account. 7. User click on confirm button.
Alternative Flow	<ol style="list-style-type: none"> 1. The method isn't valid in country <ol style="list-style-type: none"> a. Mail to the admin for the correctness and for the details. b. Admin gives information about the country currency or tell user for alternate path.

Table 54 Use Case Specification Payment Gateway Integration

3.1.19 UC Sales and Purchase

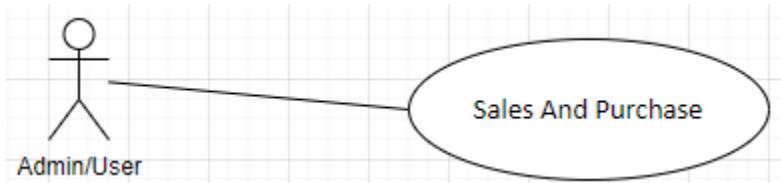


Figure 20 Use Case Sales And Purchase

Section	Detail
ID	UC-19
FR ID	FR_18, FR_19, FR_20, FR_21, FR_22
Name	Sales And Purchase
Description	It will help admin to manage sales and purchase.
Actor	Primary: Admin, User
Precondition	Sales done of the code.
Post Conditions (Success Guarantee)	User has code and developer has payment.
Basic Flow (Main Success Scenario)	<ol style="list-style-type: none"> 1. Details of the sales and purchase available to the admin. 2. Detail of the developer with payment available to the admin. 3. Detail of user with code available to the admin. 4. Admin confirms that the user has code and developer has payment.
Alternative Flow	<ol style="list-style-type: none"> 1. No option available on Administrator dashboard <ol style="list-style-type: none"> a. Admin doesn't have permission to manage sales and purchase. b. System will show the error message to admin. c. Admin will contact the Developer for permission. d. Developer will give permission to admin. 2. If given information is not valid than <ol style="list-style-type: none"> a. System will show the pop up to admin with error detail. b. Admin will check the problem with given data c. System will also show the valid information form at and pattern to admin d. Admin will flow the pattern and click again on the Detail user code button.

Table 55 Use Case Specification Sales And Purchase

3.1.20 UC Multiple Currency

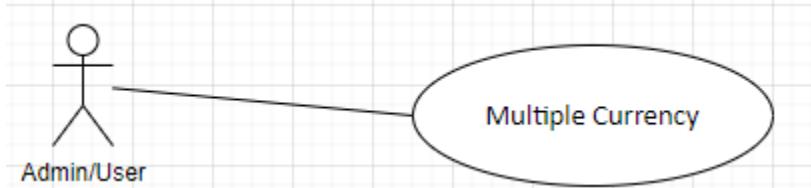


Figure 21 Use Case Multiple Currency

Section	Detail
ID	UC-20
FR ID	FR_18, FR_19, FR_20, FR_21
Name	Multiple Currency
Description	It will allow user to purchase code with multiple currency.
Actor	Primary: Admin, User
Precondition	User want to buy a code.
Post Conditions (Success Guarantee)	Payment has been done with currency chosen by user.
Basic Flow (Main Success Scenario)	<ol style="list-style-type: none"> 1. Select the code. 2. Click on the purchase code. 3. System ask for currency options. 4. User selects currency option.
Alternative Flow	<ol style="list-style-type: none"> 1. The currency isn't valid in country <ol style="list-style-type: none"> a. Mail to the admin for the correctness and for the details. b. Admin gives information about the country currency or tell user for alternate path.

Table 56 Use Case Specification Multiple Currency

3.1.21 UC Developer Sales Code



Figure 22 Use Case Developer Sales Code

Section	Detail
ID	UC-21
FR ID	FR_18, FR_19, FR_20, FR_21
Name	Developer Sales Code
Description	It will allow developer to sale his code to others.
Actor	Primary: Admin
Precondition	Developer has a code of specific problem.
Post Conditions (Success Guarantee)	Other users can see the sales.
Basic Flow (Main Success Scenario)	<ol style="list-style-type: none"> Developer shall click on the sales tab. Developer shall click on the upload code. Developer add code with problem. Developer shall click on the add price button. Add the price of code. Click on public code with price for sale.
Alternative Flow	<ol style="list-style-type: none"> Sale button is not visible <ol style="list-style-type: none"> Developer (code seller) send request to the admin. Admin check the problem and fix it Customer button is not visible <ol style="list-style-type: none"> Developer (code seller) send request to the admin Admin check the problem and fix it.

Table 57 Use Case Specification Developer Sales Code

3.1.22 UC User Purchase Code



Figure 23 Use Case User Purchase Code

Section	Detail
ID	UC-22
FR ID	FR_18, FR_19, FR_20, FR_21, FR_22
Name	User Purchase Code
Description	It will allow user to purchase a code of specific problem from the developer.
Actor	Primary: Admin, User
Precondition	Code is available on sales.
Post Conditions (Success Guarantee)	User has code and developer has payment.
Basic Flow (Main Success Scenario)	<ol style="list-style-type: none"> 1. User shall click on sales tab 2. User can search his problem on search bar. 3. User see if the solution is available. 4. User click on purchase code. 5. User make a payment method. 6. User give the payment.
Alternative Flow	<ol style="list-style-type: none"> 1. Respective code not available <ol style="list-style-type: none"> a. User send email to system admin. b. System admin will check the list of codes. c. If code isn't available than admin will add the code. d. Admin will send the confirmation email to user.

Table 58 Use Case Specification User Purchase Code

3.1.23 UC Payment Report

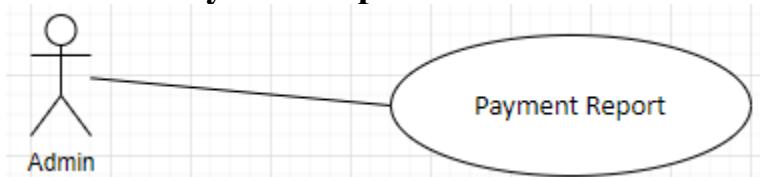


Figure 24 Use Case Payment Report

Section	Detail
ID	UC-23
FR ID	FR_18, FR_19, FR_20, FR_21, FR_22, FR_23
Name	Payment Report
Description	It will allow system to generate a payment report.
Actor	Primary: Admin
Precondition	Payment has been done by user.
Post Conditions (Success Guarantee)	Report of the payment availability.
Basic Flow (Main Success Scenario)	<ol style="list-style-type: none"> 1. Admin shall click on the sales tab. 2. Admin can view the payments. 3. Admin shall select the payment. 4. Click on the generate report button. 5. Report of the payment has been opened.
Alternative Flow	<ol style="list-style-type: none"> 1. User is not registered to the system yet <ol style="list-style-type: none"> a. Contact to the admin via email for username and password. 2. System show the error message that username or password is incorrect 3. Email Error <ol style="list-style-type: none"> a. Check the receiver email 4. File cannot upload <ol style="list-style-type: none"> a. Check the size of the file b. Check the format of the file. c. Then upload the file

Table 59 Use Case Specification Payment Report

3.1.24 UC Review on Codes

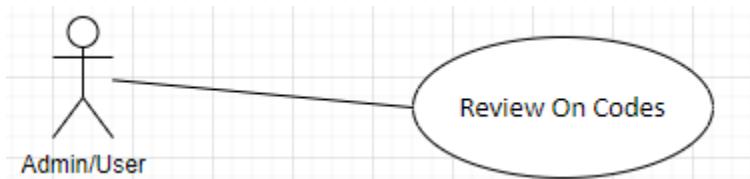


Figure 25 Use Case Review On Codes

Section	Detail
ID	UC-24
FR ID	FR_12, FR_15, FR_24, FR_29, FR_30, FR_31
Name	Review On Codes
Description	It will help the user or admin to take feedback on codes.
Actor	Primary: Admin, User
Precondition	Code availability.
Post Conditions (Success Guarantee)	User or admin has feedback on specific code.
Basic Flow (Main Success Scenario)	<ol style="list-style-type: none"> Click on the main code tab. Current code list will be open. Click on the code. Click on the comment button. Write the feedback on selected code. Click on the save button.
Alternative Flow	<ol style="list-style-type: none"> If given information is not valid than <ol style="list-style-type: none"> System will show the pop up to admin with error detail. Admin will check the problem with given data System will also show the valid information format and pattern to admin Admin will flow the pattern and click again on the comment button. Comment button is not visible <ol style="list-style-type: none"> Admin send request to the developer Developer check the problem and fix it Can't find the code <ol style="list-style-type: none"> Search for the user by its name System displays no result Admin send new record addition request to developer Developer will fix it and return Notification back to Admin

Table 60 Use Case Specification Review On Codes

3.1.25 UC Ratings

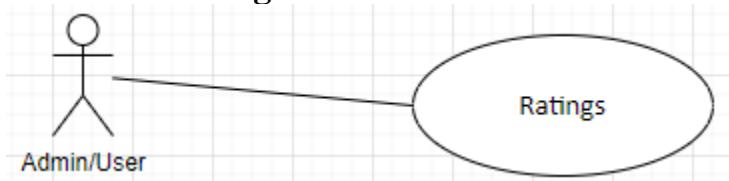


Figure 26 Use Case Ratings

Section	Detail
ID	UC-25
FR ID	FR_12, FR_15, FR_25, FR_29, FR_30, FR_31
Name	Ratings
Description	It will help the user or admin to see acceptance of codes.
Actor	Primary: Admin, User
Precondition	Code availability.
Post Conditions (Success Guarantee)	User or admin has feedback on specific code.
Basic Flow (Main Success Scenario)	<ol style="list-style-type: none"> 1. Click on the main code tab. 2. Current code list will be open. 3. Click on the code. 4. Click on the stars for rating the code. 5. Click on the save button.
Alternative Flow	<ol style="list-style-type: none"> 1. Stars for rating are not visible <ol style="list-style-type: none"> a. Admin send request to the developer b. Developer check the problem and fix it 2. Can't find the code <ol style="list-style-type: none"> a. Search for the user by its name b. System displays no result c. Admin send new record addition request to developer d. Developer will fix it and return Notification back to Admin

Table 61 Use Case Specification Ratings

3.1.26 UC Ask Query

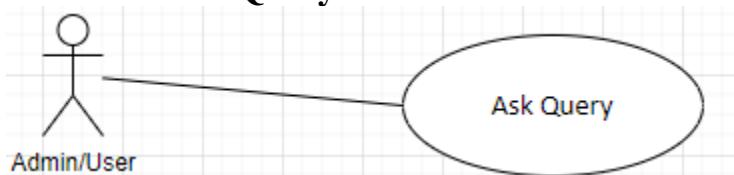


Figure 27 Use Case Ask Query

Section	Detail
ID	UC-26
FR ID	FR_07, FR_26, FR_33
Name	Ask Query
Description	It will help the users of website to ask questions from other users or from admin.
Actor	Primary: Admin, User
Precondition	User has specific query/question/problem.
Post Conditions (Success Guarantee)	Question available for the other users.
Basic Flow (Main Success Scenario)	<ol style="list-style-type: none"> 1. Click on the query tab. 2. Click on the ask question button. 3. Write the problem or query. 4. Click on done button.
Alternative Flow	<ol style="list-style-type: none"> 1. If the query is not valid <ol style="list-style-type: none"> b. Admin click on the remove query button. 2. If the query isn't in website language <ol style="list-style-type: none"> d. Admin click on the edit query button e. Change the query in appropriate words and language. f. Then click on the approve button.

Table 62 Use Case Specification Ask Query

3.1.27 UC Solution of Query

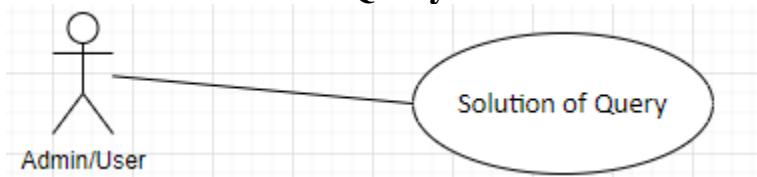


Figure 28 Use Case Solution of Query

Section	Detail
ID	UC-27
FR ID	FR_07, FR_26, FR_27, FR_33
Name	Solution of Query
Description	It will provide the solution of specific problem.
Actor	Primary: Admin, User
Precondition	Question has been asked from other users.
Post Conditions (Success Guarantee)	Solution is available for other users.
Basic Flow (Main Success Scenario)	<ol style="list-style-type: none"> Click on the query tab. Select the specific problem which solution is available. Click on the answer button. Place the solution of query. Click on the save button.
Alternative Flow	<ol style="list-style-type: none"> If given query is not valid than <ol style="list-style-type: none"> System will show the pop up to admin with error detail. Admin will check the problem with given data System will also show the valid information format and pattern to admin Admin will flow the pattern and click again on the answer button. Answer button is not visible <ol style="list-style-type: none"> Admin send request to the developer Developer check the problem and fix it Can't find the query <ol style="list-style-type: none"> Search for the user by its name System displays no result Admin send new record addition request to developer Developer will fix it and return Notification back to Admin

Table 63 Use Case Specification Solution of Query

3.1.28 UC Admin Upload Codes

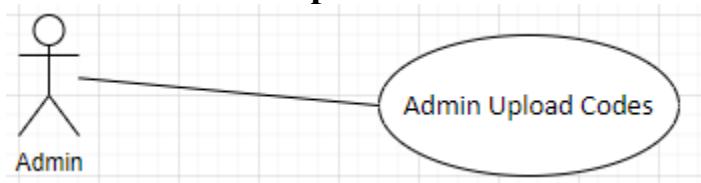


Figure 29 Use Case Admin Upload Codes

Section	Detail
ID	UC-28
FR ID	FR_12, FR_15, FR_28
Name	Admin Upload Codes
Description	Admin can upload the code on the website and description of the code will be added along code.
Actor	Primary: Admin
Precondition	Admin have code of certain requirements along description of code.
Post Conditions (Success Guarantee)	Code of specific problem available on the website.
Basic Flow (Main Success Scenario)	<ol style="list-style-type: none"> 1. Admin shall click on the main code tab. 2. Click on the add code button. 3. System shall allocate space for code. 4. Admin shall place the code on allocated space. 5. Admin click on upload button.
Alternative Flow	<ol style="list-style-type: none"> 1. Codes isn't save in database <ol style="list-style-type: none"> a. Admin send email to developer b. Developer check the problem and fix it. 2. Upload button is not visible <ol style="list-style-type: none"> a. Admin send request to the developer b. Developer check the problem and fix it. 3. Can't find the query <ol style="list-style-type: none"> a. Search for the user by its name b. System displays no result c. Admin send new record addition request to developer d. Developer will fix it and return Notification back to Admin

Table 64 Use Case Specification Admin Upload Codes

3.1.29 UC User Upload Codes

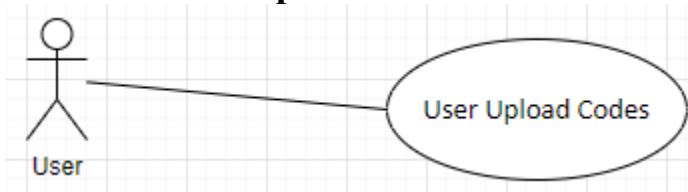


Figure 30 Use Case User Upload Codes

Section	Detail
ID	UC-29
FR ID	FR_12, FR_15, FR_29, FR_30, FR_31
Name	User Upload Codes
Description	User can upload the right code on the website and description of the code will be added.
Actor	Primary: User
Precondition	User have access on the website.
Post Conditions (Success Guarantee)	Code of specific problem available on the website.
Basic Flow (Main Success Scenario)	<ol style="list-style-type: none"> 1. User shall click on the code tab. 2. User shall click on the add code button. 3. System shall allocate space for code. 4. User shall place the code on allocated space. 5. User click on upload button.
Alternative Flow	<ol style="list-style-type: none"> 1. Codes isn't save in database <ol style="list-style-type: none"> a. User send email to Admin b. Admin check the problem and fix it. 2. Upload button is not visible <ol style="list-style-type: none"> a. User send request to the Admin. b. Admin check the problem and fix it.

Table 65 Use Case Specification User Upload Codes

3.1.30 UC User Rate Script

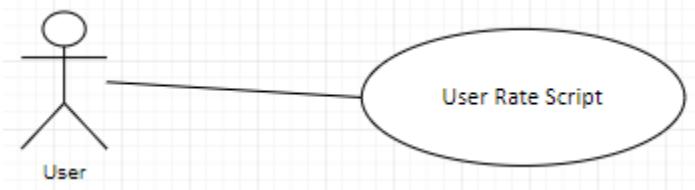


Figure 31 Use Case User Rate Script

Section	Detail
ID	UC-30
FR ID	FR_12, FR_15, FR_29, FR_30, FR_31, FR_36
Name	User Rate Script
Description	It will help user to rate a code for better.
Actor	Primary: User
Precondition	Code availability.
Post Conditions (Success Guarantee)	User rated a script of other developer.
Basic Flow (Main Success Scenario)	<ol style="list-style-type: none"> Click on the main code tab. Current script list will be open. Click on the script. Click on the stars for rating the script. Click on the save button.
Alternative Flow	<ol style="list-style-type: none"> Stars for rating are not visible <ol style="list-style-type: none"> User send request to the Admin, Admin check the problem and fix it. Can't find the code <ol style="list-style-type: none"> Search for the user by its name. System displays no result. Admin send new record addition request to developer. Developer will fix it and return Notification back to Admin.

Table 66 Use Case Specification User Rate Script

3.1.31 UC Allow User to Switch Role

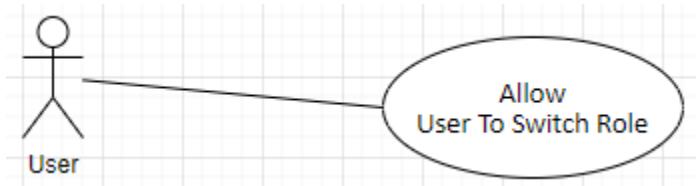


Figure 32 Use Case Allow User To Switch Role

Section	Detail
ID	UC-31
FR ID	FR_35
Name	Allow User To Switch Role
Description	It will allow user to switch their role from user to developer or from developer to user.
Actor	Primary: User
Precondition	User registered in the system.
Post Conditions (Success Guarantee)	User can work on the selected role.
Basic Flow (Main Success Scenario)	<ol style="list-style-type: none"> 1. User shall click on my details tab. 2. User shall click on role button. 3. User shall click on switch role button. 4. Select the role. 5. Role selection confirmation page will be open. 6. Click on OK button to switch role.
Alternative Flow	<ol style="list-style-type: none"> 1. Switch role button is not visible <ol style="list-style-type: none"> a. User send request to the Admin. b. Admin check the problem and fix it.

Table 67 Use Case Specification Allow User To Switch Role

3.1.32 UC User Participate In Voting

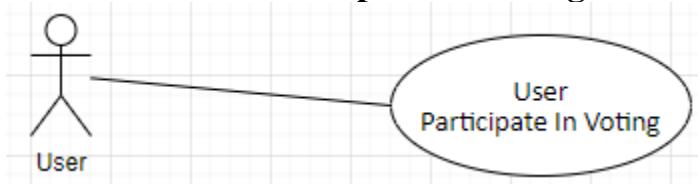


Figure 33 Use Case User Participate In Voting

Section	Detail
ID	UC-32
FR ID	FR_12, FR_15, FR_29, FR_30, FR_31, FR_34
Name	User Participate In Voting
Description	It will help user to vote for a code for code availability on system.
Actor	Primary: User
Precondition	Code available on system.
Post Conditions (Success Guarantee)	User voted a code of other developer.
Basic Flow (Main Success Scenario)	<ol style="list-style-type: none"> Click on the main code tab. Current code list will be open. Click on the code. Click on the voting code. Vote “YES” or “NO”. Click on the save button.
Alternative Flow	<ol style="list-style-type: none"> Voting button is not visible <ol style="list-style-type: none"> User send request to the Admin, Admin check the problem and fix it. Can't find the code <ol style="list-style-type: none"> Search for the user by its name. System displays no result. Admin send new record addition request to developer. Developer will fix it and return Notification back to Admin.

Table 68 Use Case Specification User Participate In Voting

3.1.33 UC Upload Script

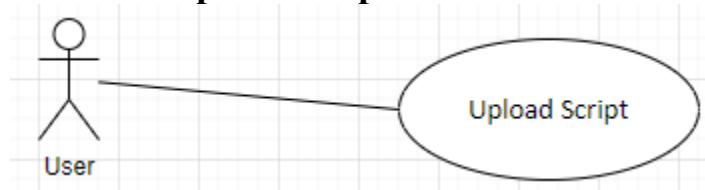


Figure 34 Use Case Upload Script

Section	Detail
ID	UC-33
FR ID	FR_12, FR_15, FR_24, FR_29, FR_30, FR_31
Name	Upload Script
Description	User can upload the script on website with their description.
Actor	Primary: User
Precondition	User have access on the website.
Post Conditions (Success Guarantee)	Script of specific problem available for others.
Basic Flow (Main Success Scenario)	<ol style="list-style-type: none"> 1. User shall click on the main code tab. 2. User shall click on the add code button. 3. System shall allocate space for script. 4. User shall place the script on allocated space. 5. User click on upload button.
Alternative Flow	<ol style="list-style-type: none"> 1. Script isn't save in database <ol style="list-style-type: none"> c. User send email to Admin d. Admin check the problem and fix it. 2. Upload button is not visible <ol style="list-style-type: none"> a. User send request to the Admin. b. Admin check the problem and fix it.

Table 69 Use Case Specification Upload Script

3.1.34 UC Delete Script

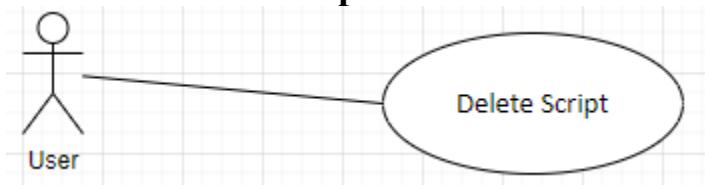


Figure 35 Use Case Delete Script

Section	Detail
ID	UC-34
FR ID	FR_12, FR_15, FR_24, FR_29, FR_30, FR_32
Name	Delete Script
Description	It will delete the script of developer.
Actor	Primary: User
Precondition	Script exists in database.
Post Conditions (Success Guarantee)	Script isn't available anymore.
Basic Flow (Main Success Scenario)	<ol style="list-style-type: none"> Click on main code tab. Current script list will be open. Select the script for deletion. Click on the delete button of the selected script. Delete confirmation page will open. Click on OK button to delete.
Alternative Flow	<ol style="list-style-type: none"> If given information is not valid than <ol style="list-style-type: none"> System will show the pop up to admin with error detail. Admin will check the problem with given data System will also show the valid information format and pattern to admin Admin will flow the pattern and click again on the Delete code button. Delete Cancelled <ol style="list-style-type: none"> The delete is cancelled, and the Basic Flow is re-started at the beginning. Deletion Error <ol style="list-style-type: none"> System will show the error message. Back to the code list.

Table 70 Use Case Specification Delete Script

3.1.35 UC Update Script

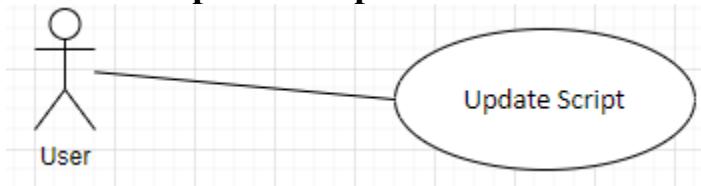


Figure 36 Use Case Update Script

Section	Detail
ID	UC-35
FR ID	FR_12, FR_15, FR_29, FR_30, FR_31, FR_31
Name	Update Script
Description	It will updated the user script.
Actor	Primary: User
Precondition	Script of the developer available on website.
Post Conditions (Success Guarantee)	Updated code is on website.
Basic Flow (Main Success Scenario)	<ol style="list-style-type: none"> Click on edit user code button. Edit developer's script. Click on the update button. Update confirmation button will be open. Click on OK for update script.
Alternative Flow	<ol style="list-style-type: none"> If the code isn't working <ol style="list-style-type: none"> Click on the edit user script button. Edit the script. Click on update button. Code details aren't added <ol style="list-style-type: none"> Compile the code and write description for it. Click on update button.

Table 71 Use Case Specification Update Script

3.1.36 UC User Ask Query

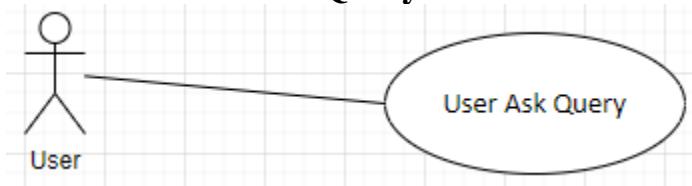


Figure 37 Use Case User Ask Query

Section	Detail
ID	UC-36
FR ID	FR_07, FR_26, FR_33
Name	User Ask Query
Description	It will help user to ask question for solution.
Actor	Primary: User
Precondition	User has a question.
Post Conditions (Success Guarantee)	Question available for the other users.
Basic Flow (Main Success Scenario)	<ol style="list-style-type: none"> Click on the query tab. Click on the ask question button. Write the problem or question. Click on done button.
Alternative Flow	<ol style="list-style-type: none"> If the query is not valid <ol style="list-style-type: none"> Admin click on the remove query button. If the query isn't in website language <ol style="list-style-type: none"> Admin click on the edit query button Change the query in appropriate words and language. Then click on the approve button.

Table 72 Use Case Specification User Ask Query

User Use Case Diagram

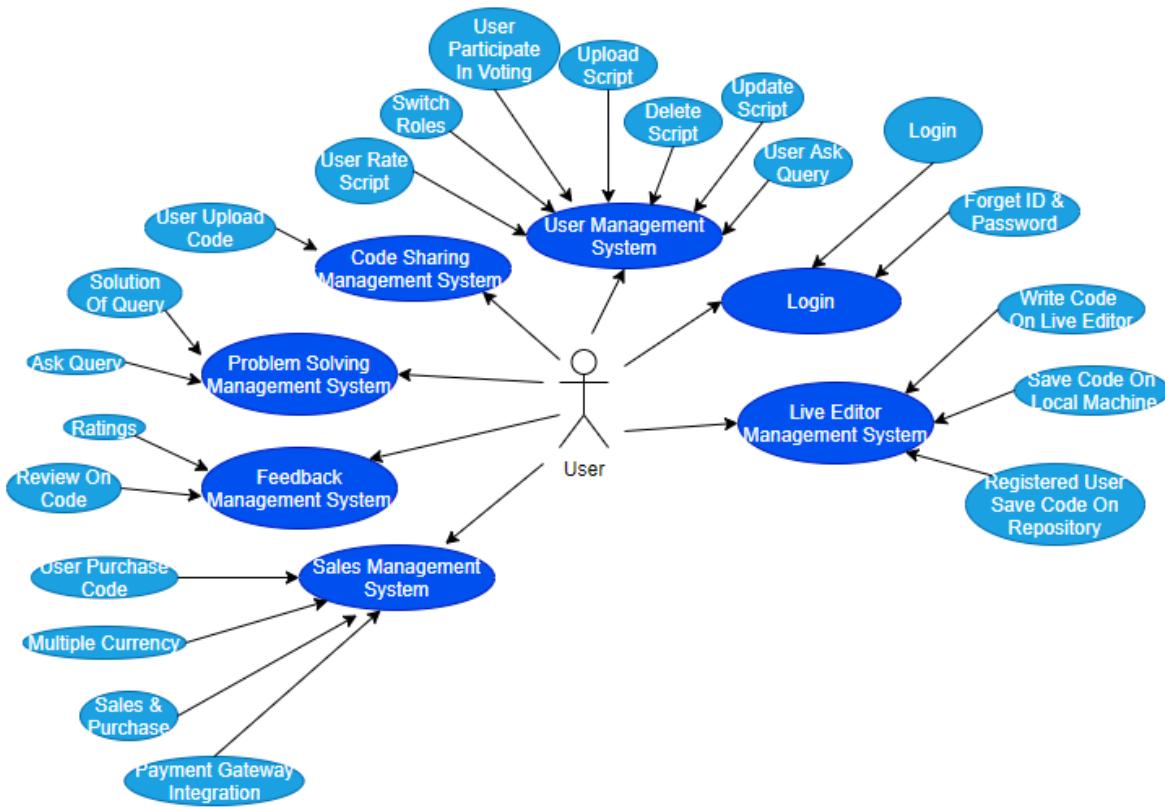


Figure 38 User Use Case Diagram

3.2 User Use Case Diagrams & Specification

3.2.1 UC Login

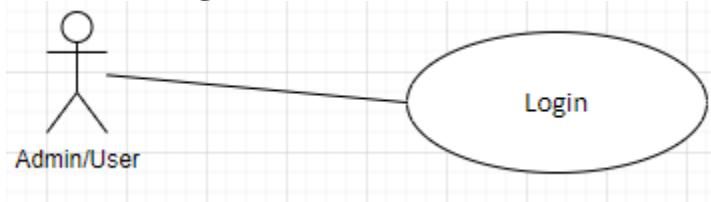


Figure 39 Use Case Login

Section	Detail
ID	UC-01
FR ID	FR_01
Name	Login
Description	It will let the user to login in the system and shows the screens according to their roles.
Actor	Primary: Admin, User
Precondition	User must have their id and password.
Post Conditions (Success Guarantee)	User logged in to the system.
Basic Flow (Main Success Scenario)	<ol style="list-style-type: none"> Enter the portal address. Login page will be open. Authenticated user will fill the login form. Submit the login form. System verifies the id and password. Dashboard of the user appears on the screen where user can perform desired tasks.
Alternative Flow	<ol style="list-style-type: none"> System fails to access portal address <ol style="list-style-type: none"> To support the system recovery, restart the system and request for the initial state. Login page is not available <ol style="list-style-type: none"> Check the internet connection and refresh the portal.

Table 73 Use Case Specification Login

3.2.2 UC Forget ID or Password

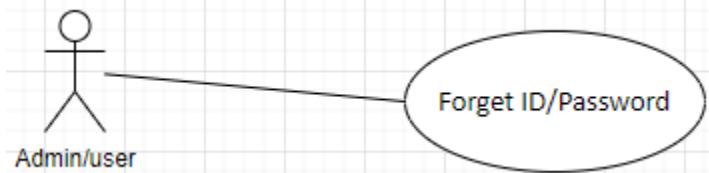


Figure 40 Use Case Forget ID or Password

Section	Detail
ID	UC-02
FR ID	FR_01, FR_02
Name	Forget ID or Password
Description	User forgets his password or id then he will recover it.
Actor	Primary: Admin, User
Precondition	User must have their ID.
Post Conditions (Success Guarantee)	The system successfully changed the login password.
Basic Flow (Main Success Scenario)	<ol style="list-style-type: none"> 1. User logged in to the system. 2. Dashboard appeared on the screen. 3. User click on User Tab. 4. User list will be appeared on the screen. 5. Select the user for reset password. 6. Click on Edit. 7. Reset the Password or Id. 8. Send Mail to User.
Alternative Flow	<ol style="list-style-type: none"> 1. User is not registered to the system yet <ol style="list-style-type: none"> a. Contact to the admin via email for username and password. 2. System show the error message that username or password is incorrect <ol style="list-style-type: none"> a. Re-enter the username and password 3. Login request is cancelled due to the poor internet connection resend the request

Table 74 Use Case Specification Forget ID or Password

3.2.3 UC Write Code on Live Editor

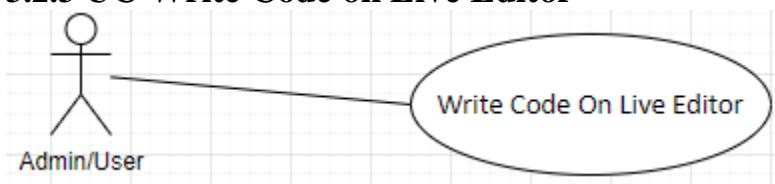


Figure 41 Use Case Write Code On Live Editor

Section	Detail
ID	UC-10
FR ID	FR_10
Name	Write Code On Live Editor
Description	It will write the code of specific problem on the live editor/compiler.
Actor	Primary: Admin, User
Precondition	Compiler available on the website.
Post Conditions (Success Guarantee)	Code with the description and output has been wrote on compiler.
Basic Flow (Main Success Scenario)	<ol style="list-style-type: none"> Click on the compiler tab on the website. Select the languages available. Write the code on the compiler. Outputs has been shown at that moment.
Alternative Flow	<ol style="list-style-type: none"> System fails to access portal address <ol style="list-style-type: none"> To support the system recovery, restart the system and request for the initial state. Live editor page is not available <ol style="list-style-type: none"> Check the internet connection and refresh the portal.

Table 75 Use Case Specification Write Code On Live Editor

3.2.4 UC Save Code on Local Machine

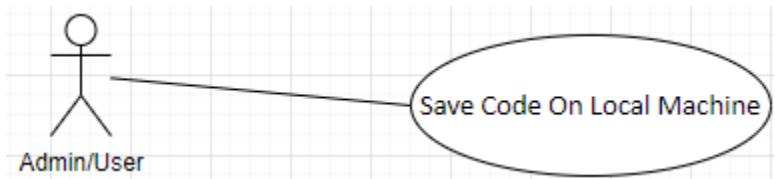


Figure 42 Use Case Save Code On Local Machine

Section	Detail
ID	UC-11
FR ID	FR_10, FR_11
Name	Save Code On Local Machine
Description	It will save the code on compiler temporary.
Actor	Primary: Admin, User
Precondition	Code has been written on live editor.
Post Conditions (Success Guarantee)	Code successfully saved on local machine/compiler.
Basic Flow (Main Success Scenario)	<ol style="list-style-type: none"> Click on the save button. Click on the local machine tab.
Alternative Flow	<ol style="list-style-type: none"> Live editor page is not working properly <ol style="list-style-type: none"> Check the internet connection and refresh the portal.

Table 76 Use Case Specification Save Code On Local Machine

3.2.5 UC Allow Registered User to Save Code In Repository

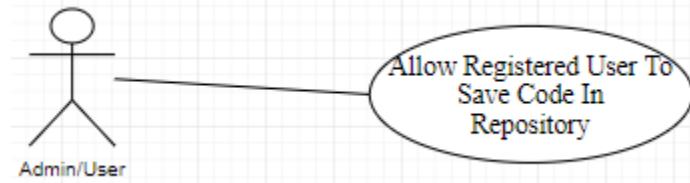


Figure 43 Use Case Allow Registered User To Save Code In Repository

Section	Detail
ID	UC-12
FR ID	FR_10, FR_11, FR_12
Name	Allow Registered User To Save Code In Repository
Description	It will save the code to database from compiler.
Actor	Primary: Admin, User
Precondition	Code has been written on the live editor.
Post Conditions (Success Guarantee)	Code successfully saved on to database from local machine.
Basic Flow (Main Success Scenario)	<ol style="list-style-type: none"> Click on the save as button. Options will be appear on the screen. Select the desire option. Click on the database. Click on the confirm button.
Alternative Flow	<ol style="list-style-type: none"> If the code isn't correct <ol style="list-style-type: none"> A pop up to user to change his code or correct it. If code is already available on system <ol style="list-style-type: none"> A pop up to user that code is already available on database.

Table 77 Use Case Specification Allow Registered User To Save Code In Repository

3.2.6 UC Payment Gateway Integration

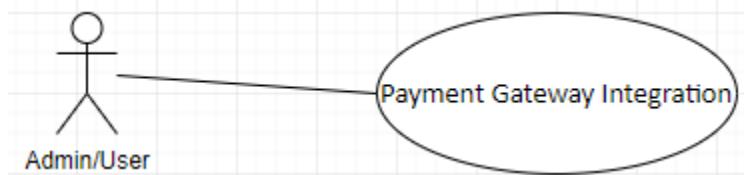


Figure 44 Use Case Payment Gateway Integration

Section	Detail
ID	UC-18
FR ID	FR_18, FR_19, FR_20, FR_21
Name	Payment Gateway Integration
Description	It will allow user to purchase code with multiple methods available in country.
Actor	Primary: Admin, User
Precondition	Payment option available.
Post Conditions (Success Guarantee)	Payment has been done with currency chosen by user.
Basic Flow (Main Success Scenario)	<ol style="list-style-type: none"> 1. Select the code. 2. Click on the purchase code. 3. System ask for payment method options. 4. User selects method. 5. System ask for account option. 6. User enter his available account. 7. User click on confirm button.
Alternative Flow	<ol style="list-style-type: none"> 1. The method isn't valid in country <ol style="list-style-type: none"> a. Mail to the admin for the correctness and for the details. b. Admin gives information about the country currency or tell user for alternate path.

Table 78 Use Case Specification Payment Gateway Integration

3.2.7 UC Sales and Purchase

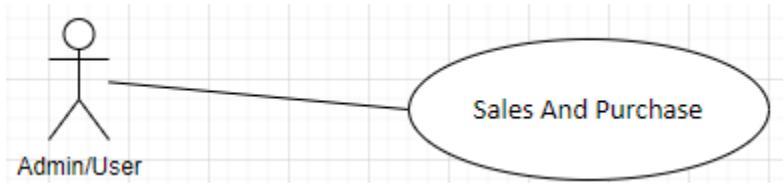


Figure 45 Use Case Sales And Purchase

Section	Detail
ID	UC-19
FR ID	FR_18, FR_19, FR_20, FR_21, FR_22
Name	Sales And Purchase
Description	It will help admin to manage sales and purchase.
Actor	Primary: Admin, User
Precondition	Sales done of the code.
Post Conditions (Success Guarantee)	User has code and developer has payment.
Basic Flow (Main Success Scenario)	<ol style="list-style-type: none"> 1. Details of the sales and purchase available to the admin. 2. Detail of the developer with payment available to the admin. 3. Detail of user with code available to the admin. 4. Admin confirms that the user has code and developer has payment.
Alternative Flow	<ol style="list-style-type: none"> 1. No option available on Administrator dashboard <ol style="list-style-type: none"> a. Admin doesn't have permission to manage sales and purchase. b. System will show the error message to admin. c. Admin will contact the Developer for permission. d. Developer will give permission to admin. 2. If given information is not valid than <ol style="list-style-type: none"> a. System will show the pop up to admin with error detail. b. Admin will check the problem with given data c. System will also show the valid information form at and pattern to admin d. Admin will flow the pattern and click again on the Detail user code button.

Table 79 Use Case Specification Sales And Purchase

3.2.8 UC Multiple Currency

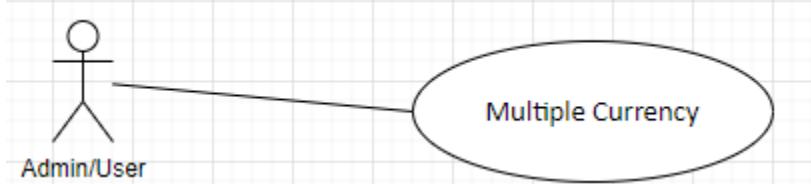


Figure 46 Use Case Multiple Currency

Section	Detail
ID	UC-20
FR ID	FR_18, FR_19, FR_20, FR_21
Name	Multiple Currency
Description	It will allow user to purchase code with multiple currency.
Actor	Primary: Admin, User
Precondition	User want to buy a code.
Post Conditions (Success Guarantee)	Payment has been done with currency chosen by user.
Basic Flow (Main Success Scenario)	<ol style="list-style-type: none"> 1. Select the code. 2. Click on the purchase code. 3. System ask for currency options. 4. User selects currency option.
Alternative Flow	<ol style="list-style-type: none"> 1. The currency isn't valid in country <ol style="list-style-type: none"> a. Mail to the admin for the correctness and for the details. b. Admin gives information about the country currency or tell user for alternate path.

Table 80 Use Case Specification Multiple Currency

3.2.9 UC User Purchase Code

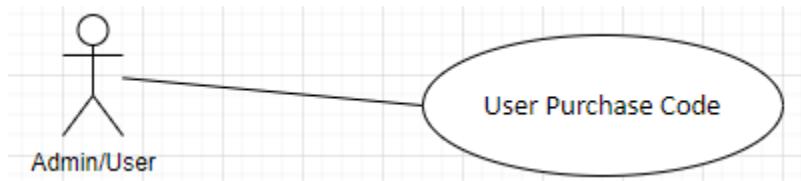


Figure 47 Use Case User Purchase Code

Section	Detail
ID	UC-22
FR ID	FR_18, FR_19, FR_20, FR_21, FR_22
Name	User Purchase Code
Description	It will allow user to purchase a code of specific problem from the developer.
Actor	Primary: Admin, User
Precondition	Code is available on sales.
Post Conditions (Success Guarantee)	User has code and developer has payment.
Basic Flow (Main Success Scenario)	<ol style="list-style-type: none"> 1. User shall click on sales tab 2. User can search his problem on search bar. 3. User see if the solution is available. 4. User click on purchase code. 5. User make a payment method. 6. User give the payment.
Alternative Flow	<ol style="list-style-type: none"> 1. Respective code not available <ol style="list-style-type: none"> a. User send email to system admin. b. System admin will check the list of codes. c. If code isn't available than admin will add the code. d. Admin will send the confirmation email to user.

Table 81 Use Case Specification User Purchase Code

3.2.10 UC Review on Codes

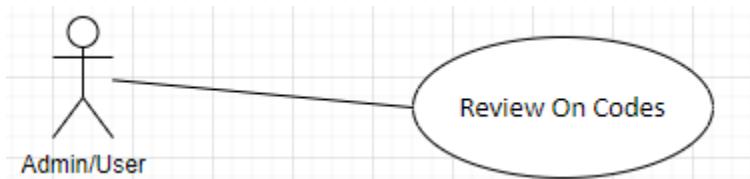


Figure 48 Use Case Review On Codes

Section	Detail
ID	UC-24
FR ID	FR_12, FR_15, FR_24, FR_29, FR_30, FR_31
Name	Review On Codes
Description	It will help the user or admin to take feedback on codes.
Actor	Primary: Admin, User
Precondition	Code availability.
Post Conditions (Success Guarantee)	User or admin has feedback on specific code.
Basic Flow (Main Success Scenario)	<ol style="list-style-type: none"> Click on the main code tab. Current code list will be open. Click on the code. Click on the comment button. Write the feedback on selected code. Click on the save button.
Alternative Flow	<ol style="list-style-type: none"> If given information is not valid than <ol style="list-style-type: none"> System will show the pop up to admin with error detail. Admin will check the problem with given data System will also show the valid information format and pattern to admin Admin will flow the pattern and click again on the comment button. Comment button is not visible <ol style="list-style-type: none"> Admin send request to the developer Developer check the problem and fix it Can't find the code <ol style="list-style-type: none"> Search for the user by its name System displays no result Admin send new record addition request to developer Developer will fix it and return Notification back to Admin

Table 82 Use Case Specification Review On Codes

3.2.11 UC Ratings

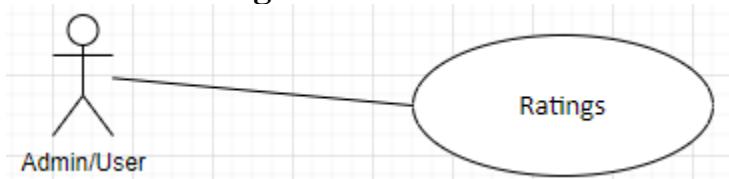


Figure 49 Use Case Ratings

Section	Detail
ID	UC-25
FR ID	FR_12, FR_15, FR_25, FR_29, FR_30, FR_31
Name	Ratings
Description	It will help the user or admin to see acceptance of codes.
Actor	Primary: Admin, User
Precondition	Code availability.
Post Conditions (Success Guarantee)	User or admin has feedback on specific code.
Basic Flow (Main Success Scenario)	<ol style="list-style-type: none"> Click on the main code tab. Current code list will be open. Click on the code. Click on the stars for rating the code. Click on the save button.
Alternative Flow	<ol style="list-style-type: none"> Stars for rating are not visible <ol style="list-style-type: none"> Admin send request to the developer Developer check the problem and fix it Can't find the code <ol style="list-style-type: none"> Search for the user by its name System displays no result Admin send new record addition request to developer Developer will fix it and return Notification back to Admin

Table 83 Use Case Specification Ratings

3.2.12 UC Ask Query

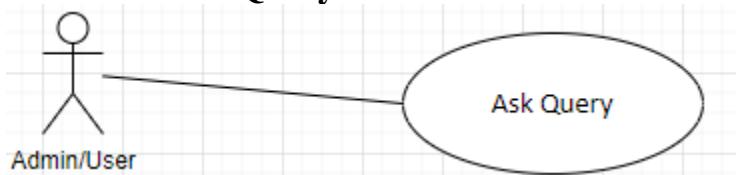


Figure 50 Use Case Ask Query

Section	Detail
ID	UC-26
FR ID	FR_07, FR_26, FR_33
Name	Ask Query
Description	It will help the users of website to ask questions from other users or from admin.
Actor	Primary: Admin, User
Precondition	User has specific query/question/problem.
Post Conditions (Success Guarantee)	Question available for the other users.
Basic Flow (Main Success Scenario)	<ol style="list-style-type: none"> Click on the query tab. Click on the ask question button. Write the problem or query. Click on done button.
Alternative Flow	<ol style="list-style-type: none"> If the query is not valid <ol style="list-style-type: none"> Admin click on the remove query button. If the query isn't in website language <ol style="list-style-type: none"> Admin click on the edit query button Change the query in appropriate words and language. Then click on the approve button.

Table 84 Use Case Specification Ask Query

3.2.13 UC Solution of Query

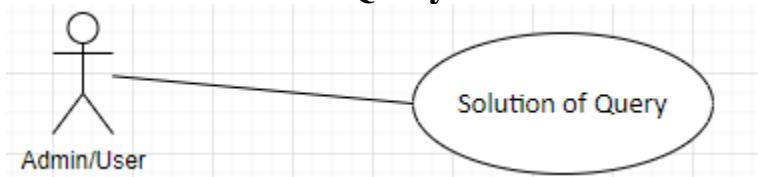


Figure 51 Use Case Solution of Query

Section	Detail
ID	UC-27
FR ID	FR_07, FR_26, FR_27, FR_33
Name	Solution of Query
Description	It will provide the solution of specific problem.
Actor	Primary: Admin, User
Precondition	Question has been asked from other users.
Post Conditions (Success Guarantee)	Solution is available for other users.
Basic Flow (Main Success Scenario)	<ol style="list-style-type: none"> Click on the query tab. Select the specific problem which solution is available. Click on the answer button. Place the solution of query. Click on the save button.
Alternative Flow	<ol style="list-style-type: none"> If given query is not valid than <ol style="list-style-type: none"> System will show the pop up to admin with error detail. Admin will check the problem with given data System will also show the valid information format and pattern to admin Admin will flow the pattern and click again on the answer button. Answer button is not visible <ol style="list-style-type: none"> Admin send request to the developer Developer check the problem and fix it Can't find the query <ol style="list-style-type: none"> Search for the user by its name System displays no result Admin send new record addition request to developer Developer will fix it and return Notification back to Admin

Table 85 Use Case Specification Solution of Query

3.2.14 UC User Upload Codes

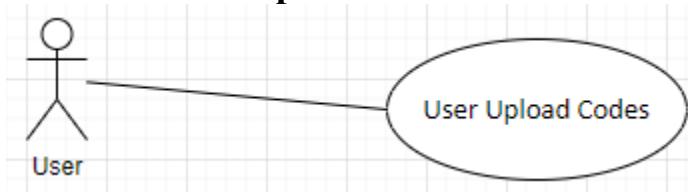


Figure 52 Use Case User Upload Codes

Section	Detail
ID	UC-29
FR ID	FR_12, FR_15, FR_29, FR_30, FR_31
Name	User Upload Codes
Description	User can upload the right code on the website and description of the code will be added.
Actor	Primary: User
Precondition	User have access on the website.
Post Conditions (Success Guarantee)	Code of specific problem available on the website.
Basic Flow (Main Success Scenario)	<ol style="list-style-type: none"> 1. User shall click on the code tab. 2. User shall click on the add code button. 3. System shall allocate space for code. 4. User shall place the code on allocated space. 5. User click on upload button.
Alternative Flow	<ol style="list-style-type: none"> 1. Codes isn't save in database <ol style="list-style-type: none"> a. User send email to Admin b. Admin check the problem and fix it. 2. Upload button is not visible <ol style="list-style-type: none"> a. User send request to the Admin. b. Admin check the problem and fix it.

Table 86 Use Case Specification User Upload Codes

3.2.15 UC User Rate Script

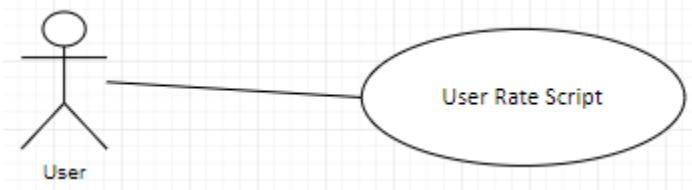


Figure 53 Use Case User Rate Script

Section	Detail
ID	UC-30
FR ID	FR_12, FR_15, FR_29, FR_30, FR_31, FR_36
Name	User Rate Script
Description	It will help user to rate a code for better.
Actor	Primary: User
Precondition	Code availability.
Post Conditions (Success Guarantee)	User rated a script of other developer.
Basic Flow (Main Success Scenario)	<ol style="list-style-type: none"> Click on the main code tab. Current script list will be open. Click on the script. Click on the stars for rating the script. Click on the save button.
Alternative Flow	<ol style="list-style-type: none"> Stars for rating are not visible <ol style="list-style-type: none"> User send request to the Admin, Admin check the problem and fix it. Can't find the code <ol style="list-style-type: none"> Search for the user by its name. System displays no result. Admin send new record addition request to developer. Developer will fix it and return Notification back to Admin.

Table 87 Use Case Specification User Rate Script

3.2.16 UC Allow User To Switch Role

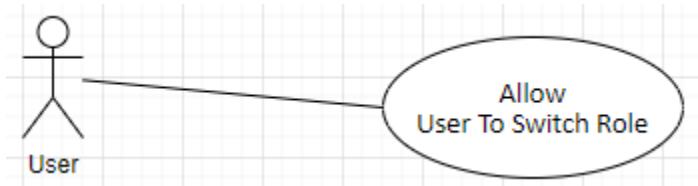


Figure 54 Use Case Allow User To Switch Role

Section	Detail
ID	UC-31
FR ID	FR_35
Name	Allow User To Switch Role
Description	It will allow user to switch their role from user to developer or from developer to user.
Actor	Primary: User
Precondition	User registered in the system.
Post Conditions (Success Guarantee)	User can work on the selected role.
Basic Flow (Main Success Scenario)	<ol style="list-style-type: none"> 1. User shall click on my details tab. 2. User shall click on role button. 3. User shall click on switch role button. 4. Select the role. 5. Role selection confirmation page will be open. 6. Click on OK button to switch role.
Alternative Flow	<ol style="list-style-type: none"> 1. Switch role button is not visible <ol style="list-style-type: none"> a. User send request to the Admin. b. Admin check the problem and fix it.

Table 88 Use Case Specification Allow User To Switch Role

3.2.17 UC User Participate In Voting

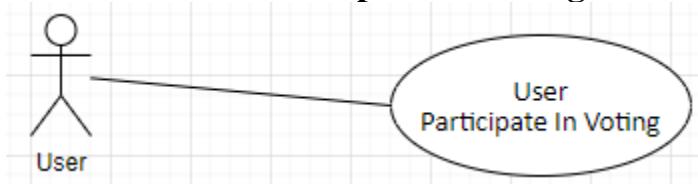


Figure 55 Use Case User Participate In Voting

Section	Detail
ID	UC-32
FR ID	FR_12, FR_15, FR_29, FR_30, FR_31, FR_34
Name	User Participate In Voting
Description	It will help user to vote for a code for code availability on system.
Actor	Primary: User
Precondition	Code available on system.
Post Conditions (Success Guarantee)	User voted a code of other developer.
Basic Flow (Main Success Scenario)	<ol style="list-style-type: none"> Click on the main code tab. Current code list will be open. Click on the code. Click on the voting code. Vote “YES” or “NO”. Click on the save button.
Alternative Flow	<ol style="list-style-type: none"> Voting button is not visible <ol style="list-style-type: none"> User send request to the Admin, Admin check the problem and fix it. Can't find the code <ol style="list-style-type: none"> Search for the user by its name. System displays no result. Admin send new record addition request to developer. Developer will fix it and return Notification back to Admin.

Table 89 Use Case Specification User Participate In Voting

3.2.18 UC Upload Script

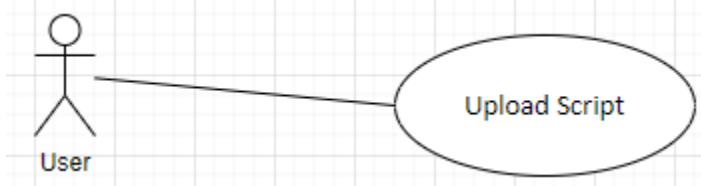


Figure 56 Use Case Upload Script

Section	Detail
ID	UC-33
FR ID	FR_12, FR_15, FR_24, FR_29, FR_30, FR_31
Name	Upload Script
Description	User can upload the script on website with their description.
Actor	Primary: User
Precondition	User have access on the website.
Post Conditions (Success Guarantee)	Script of specific problem available for others.
Basic Flow (Main Success Scenario)	<ol style="list-style-type: none"> 1. User shall click on the main code tab. 2. User shall click on the add code button. 3. System shall allocate space for script. 4. User shall place the script on allocated space. 5. User click on upload button.
Alternative Flow	<ol style="list-style-type: none"> 1. Script isn't save in database <ol style="list-style-type: none"> a. User send email to Admin b. Admin check the problem and fix it. 2. Upload button is not visible <ol style="list-style-type: none"> a. User send request to the Admin. b. Admin check the problem and fix it.

Table 90 Use Case Specification Upload Script

3.2.19 UC Delete Script

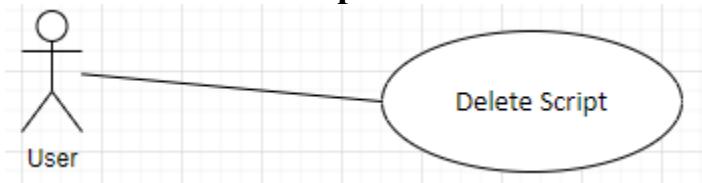


Figure 57 Use Case Delete Script

Section	Detail
ID	UC-34
FR ID	FR_12, FR_15, FR_24, FR_29, FR_30, FR_32
Name	Delete Script
Description	It will delete the script of developer.
Actor	Primary: User
Precondition	Script exists in database.
Post Conditions (Success Guarantee)	Script isn't available anymore.
Basic Flow (Main Success Scenario)	<ol style="list-style-type: none"> Click on main code tab. Current script list will be open. Select the script for deletion. Click on the delete button of the selected script. Delete confirmation page will open. Click on OK button to delete.
Alternative Flow	<ol style="list-style-type: none"> If given information is not valid than <ol style="list-style-type: none"> System will show the pop up to admin with error detail. Admin will check the problem with given data System will also show the valid information format and pattern to admin Admin will flow the pattern and click again on the Delete code button. Delete Cancelled <ol style="list-style-type: none"> The delete is cancelled, and the Basic Flow is re-started at the beginning. Deletion Error <ol style="list-style-type: none"> System will show the error message. Back to the code list.

Table 91 Use Case Specification Delete Script

3.2.20 UC Update Script

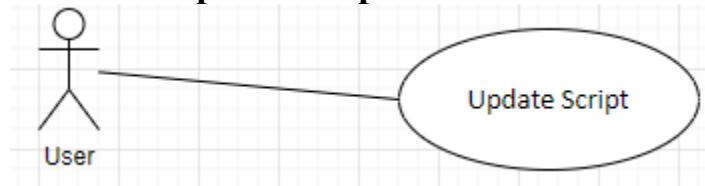


Figure 58 Use Case Update Script

Section	Detail
ID	UC-35
FR ID	FR_12, FR_15, FR_29, FR_30, FR_31, FR_31
Name	Update Script
Description	It will updated the user script.
Actor	Primary: User
Precondition	Script of the developer available on website.
Post Conditions (Success Guarantee)	Updated code is on website.
Basic Flow (Main Success Scenario)	<ol style="list-style-type: none"> Click on edit user code button. Edit developer's script. Click on the update button. Update confirmation button will be open. Click on OK for update script.
Alternative Flow	<ol style="list-style-type: none"> If the code isn't working <ol style="list-style-type: none"> Click on the edit user script button. Edit the script. Click on update button. Code details aren't added <ol style="list-style-type: none"> Compile the code and write description for it. Click on update button.

Table 92 Use Case Specification Update Script

3.2.21 UC User Ask Query

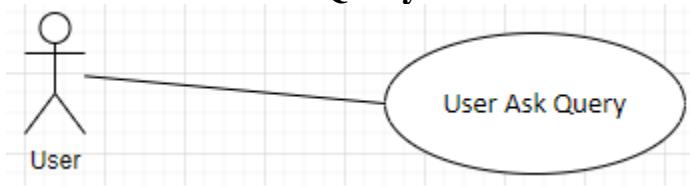


Figure 59 Use Case User Ask Query

Section	Detail
ID	UC-36
FR ID	FR_07, FR_26, FR_33
Name	User Ask Query
Description	It will help user to ask question for solution.
Actor	Primary: User
Precondition	User has a question.
Post Conditions (Success Guarantee)	Question available for the other users.
Basic Flow (Main Success Scenario)	<ol style="list-style-type: none"> Click on the query tab. Click on the ask question button. Write the problem or question. Click on done button.
Alternative Flow	<ol style="list-style-type: none"> If the query is not valid <ol style="list-style-type: none"> Admin click on the remove query button. If the query isn't in website language <ol style="list-style-type: none"> Admin click on the edit query button Change the query in appropriate words and language. Then click on the approve button.

Table 93 Use Case Specification User Ask Query

Chapter 4

Design

4.1 Architecture Diagram

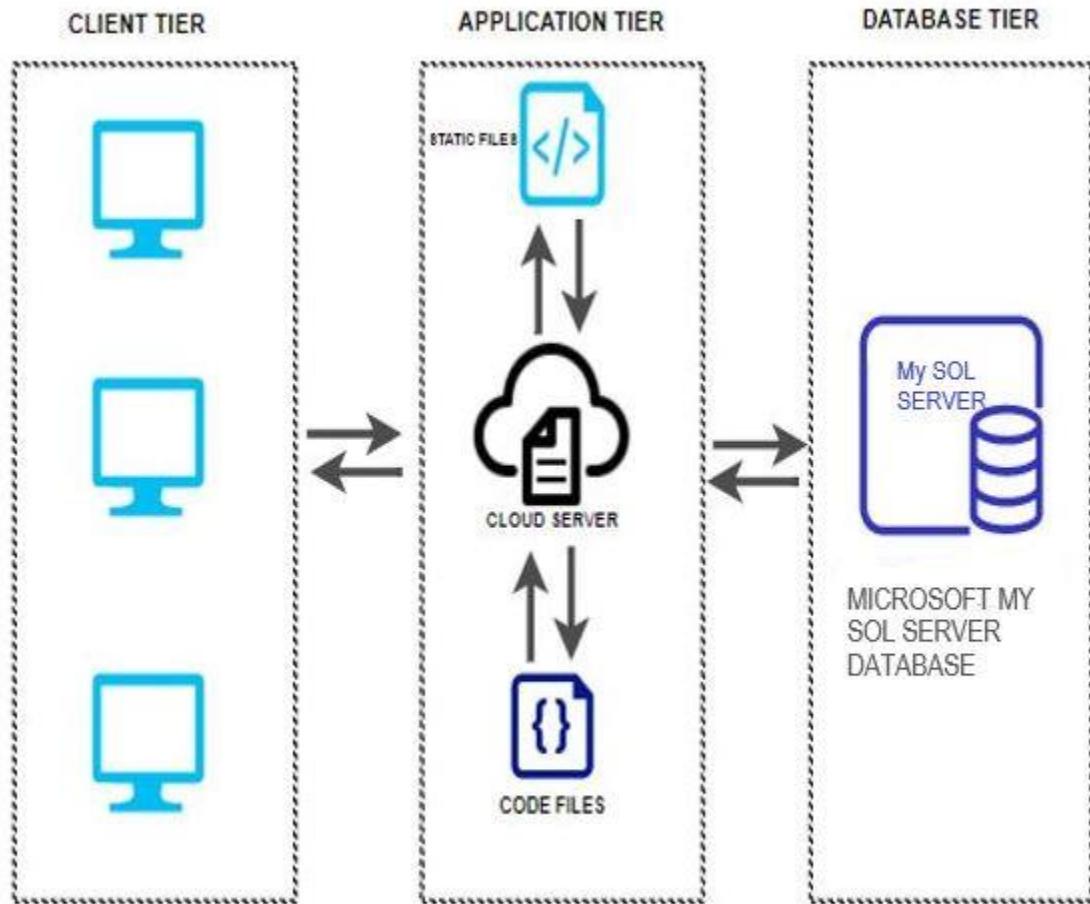


Figure 60 Architecture Diagram

4.2 Entity Relationship Diagram with Data Dictionary

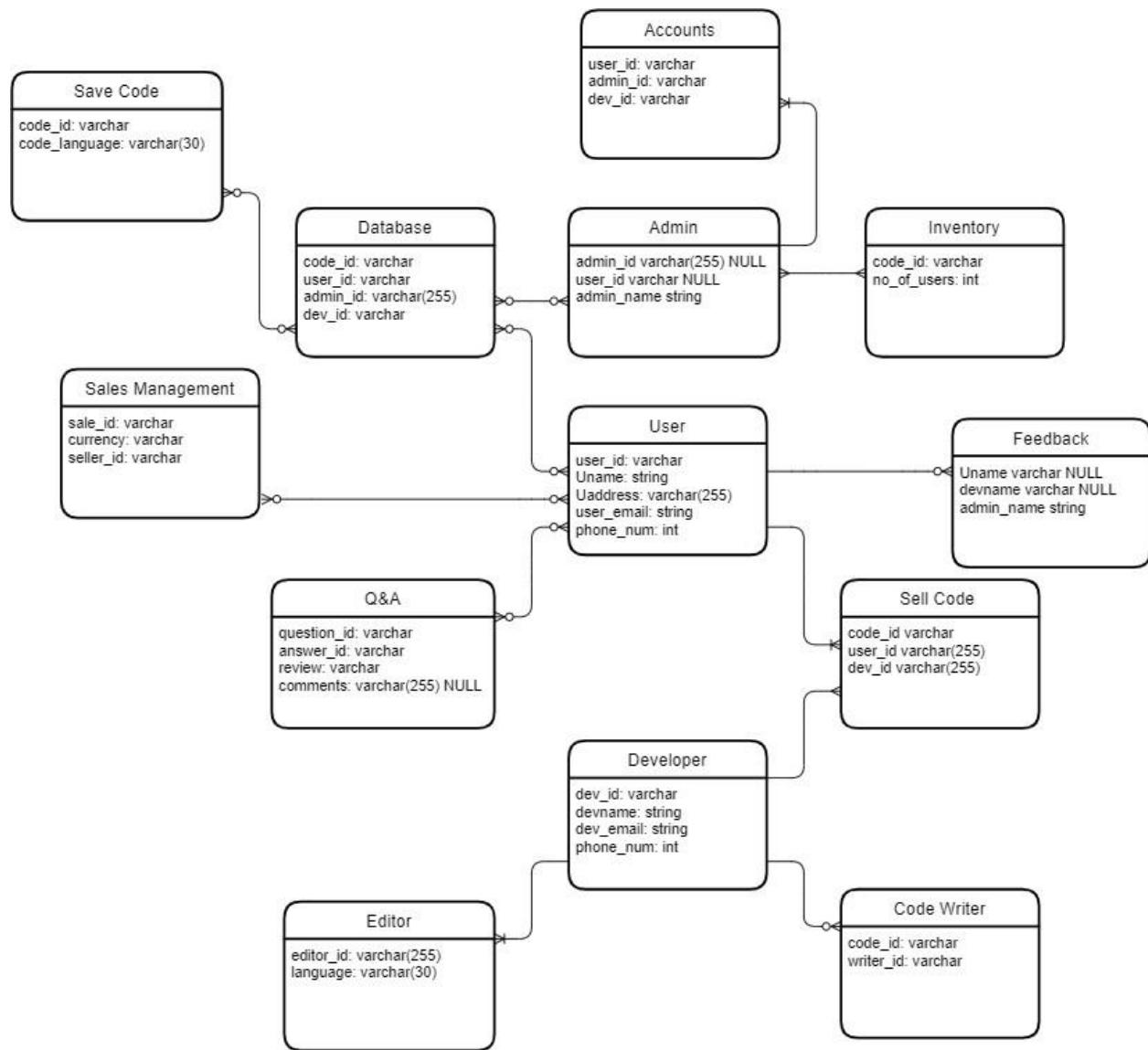


Figure 61 Entity Relationship Diagram with Data Dictionary

Data Dictionary against ERD Diagram

User

Name	Data Type	Description
user_id		Primary key
Uname	string	Attribute
Uaddress	varchar(255)	Attribute
User_email	string	Attribute
phone_num	int	Attribute

Table 94 Data dictionary against ERD for user

Admin

Name	Data Type	Description
admin_id	varchar(255)	Primary key
user_id	varchar(255)	Foreign key
admin_name	string	Attribute

Table 95 Data dictionary against ERD for admin

Accounts

Name	Data Type	Description
user_id	varchar	Foreign key
admin_id	varchar	Foreign key
dev_id	varchar	Primary key

Table 96 Data dictionary against ERD for accounts

Database

Name	Data Type	Description
code_id	varchar(255)	Foreign key
user_id	varchar(255)	Foreign key
admin_id	varchar(255)	Foreign key
dev_id	varchar(255)	Primary key

Table 97 Data dictionary against ERD for database

Save Code

Name	Data Type	Description
code_id	vachar	Foreign key
code_language	Varchar(30)	Attribute

Table 98 Data dictionary against ERD for save code

Inventory

Name	Data Type	Description
code_id	varchar	Foreign key
no_of_users	int	Attribute

Table 99 Data dictionary against ERD for inventory

Sales Management

Name	Data Type	Description
sale_id	varchar(255)	Primary key
currency	string	Attribute
seller_id	varchar(100)	Primary key

Table 100 Data dictionary against ERD for sales management

Feedback

Name	Data Type	Description
Uname	string	Attribute
dev_name	string	Attribute
admin_name	string	Attribute

Table 101 Data dictionary against ERD for feedback

Q&A

Name	Data Type	Description
question_id	varchar	Primary key
answer_id	varchar	Primary key
review	varchar	Attribute
comments	string	Attribute

Table 102 Data dictionary against ERD for Q&A

Sell Code

Name	Data Type	Description
code_id	varchar	Primary key
user_id	varchar(255)	Foreign key
dev_id	varchar(255)	Foreign key

Table 103 Data dictionary against ERD for Sell Code

Developer

Name	Data Type	Description
dev_id	varchar	Primary Key
dev_name	string	Attribute
dev_email	string	Attribute
phone_num	int	Attribute

Table 104 Data dictionary against ERD for Developer

Editor

Name	Data Type	Description
editor_id	varchar(255)	Primary Key
language	string	Attribute

Table 105 Data dictionary against ERD for Editor

Code Writer

Name	Data Type	Description
code_id	varchar	Foreign key
writer_id	varchar	Attribute

Table 106 Data Dictionary against ERD for code writer

4.3 Data Flow Diagram

4.3.1 Level 0

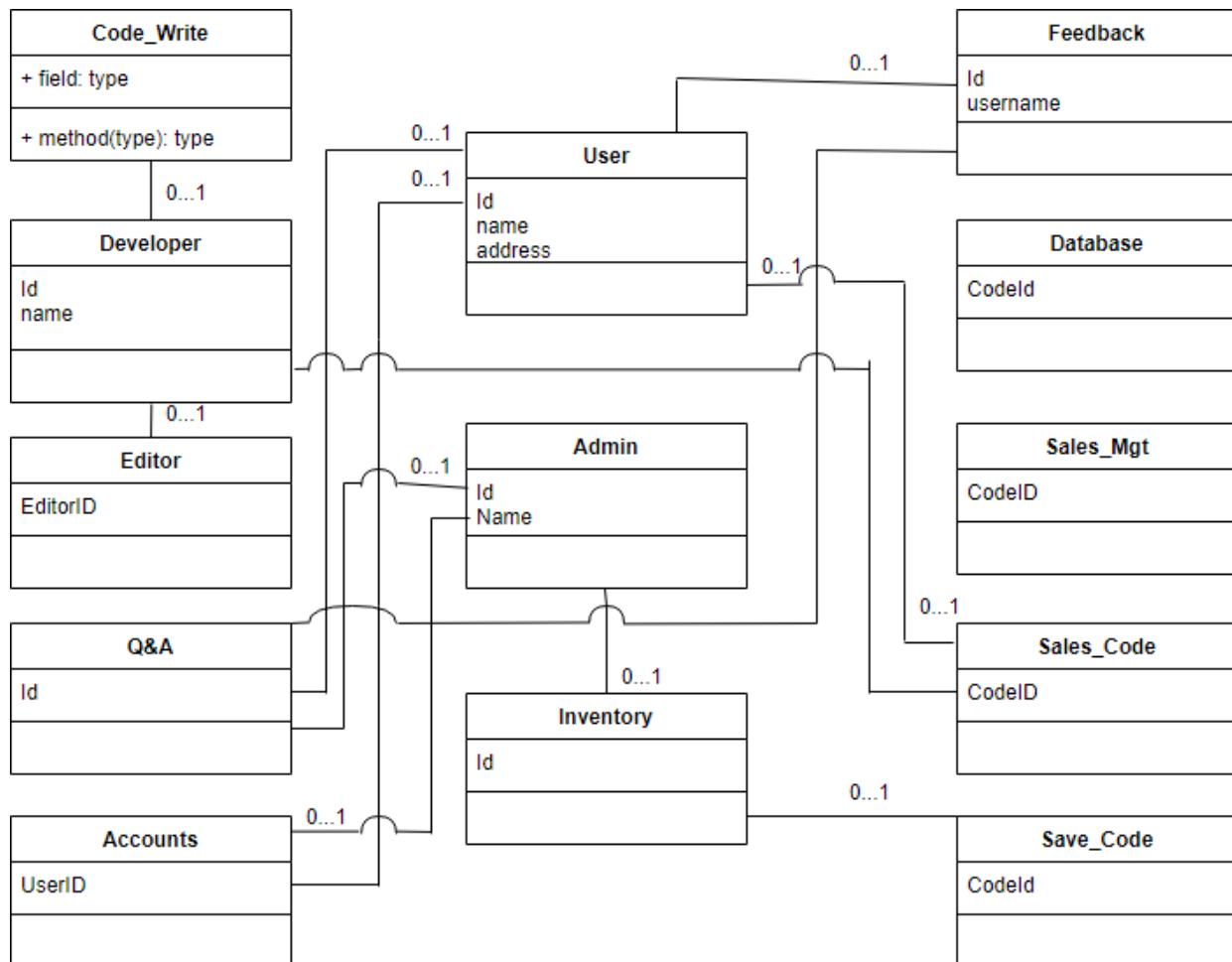


Figure 62 Data Flow Diagram Level 0

4.3.1 Level 1

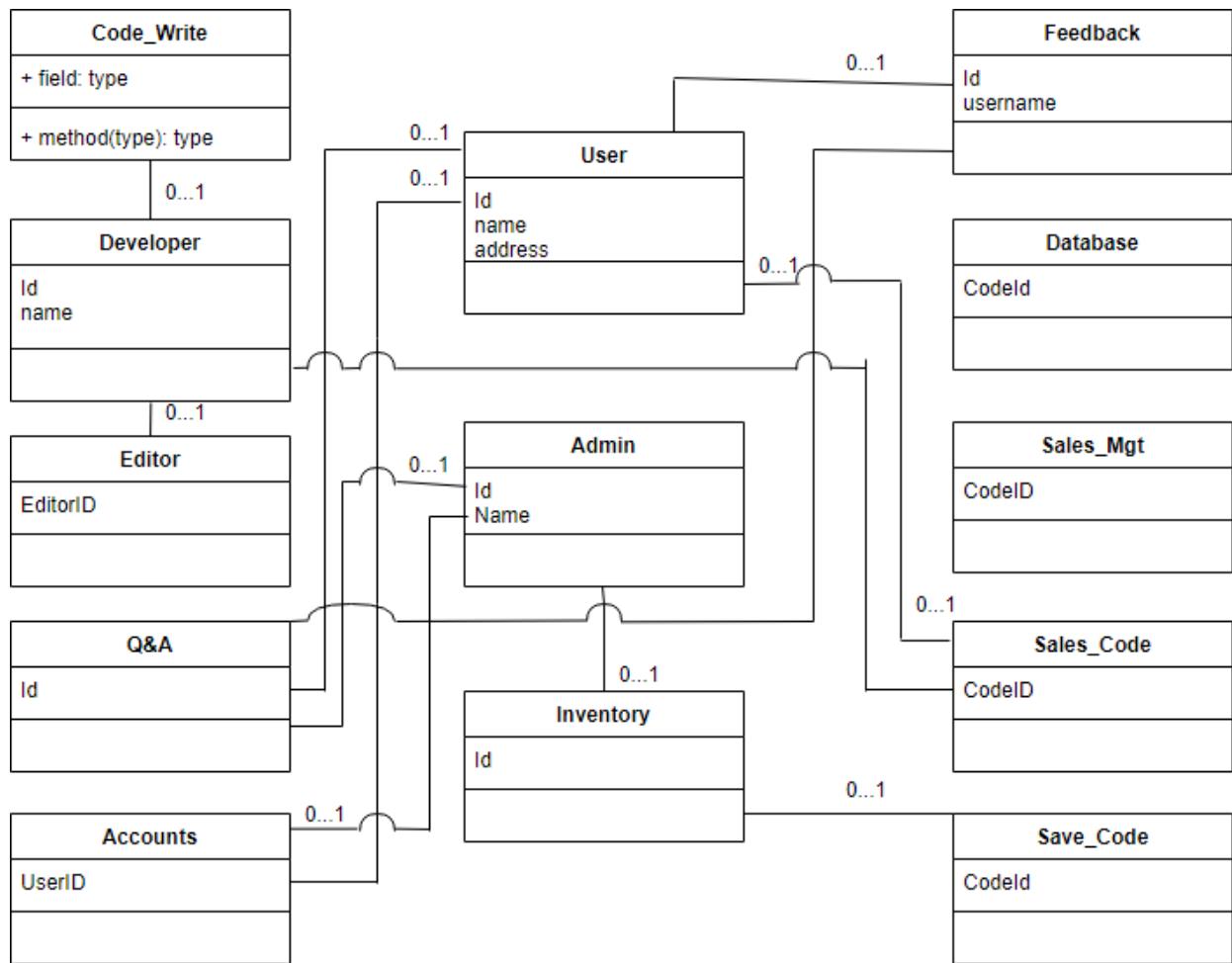


Figure 63 Data Flow Diagram Level 1

4.4 Class Diagram

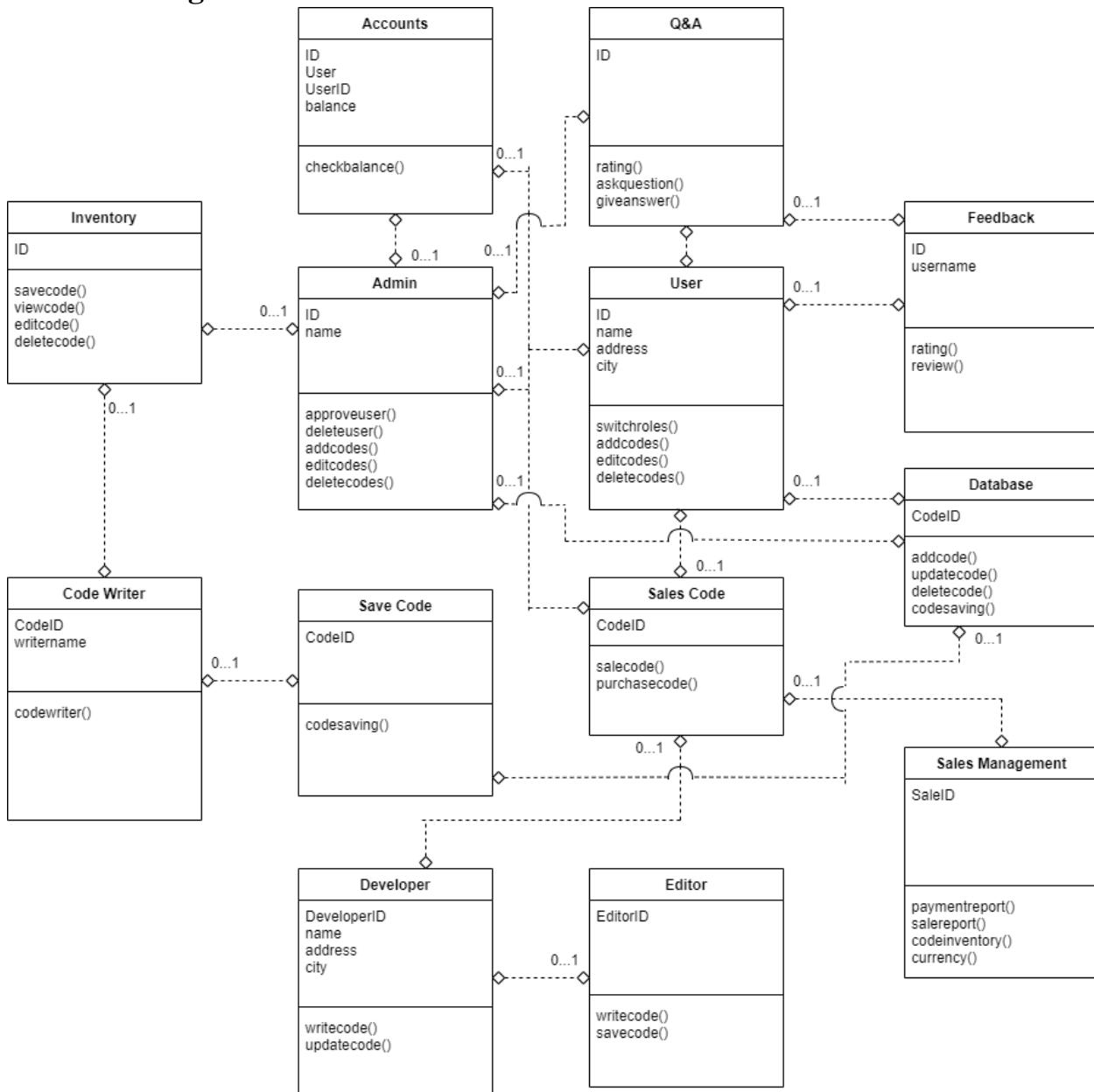


Figure 64 Class Diagram

4.5 Activity Diagram

Admin Panel

4.5.1 Login

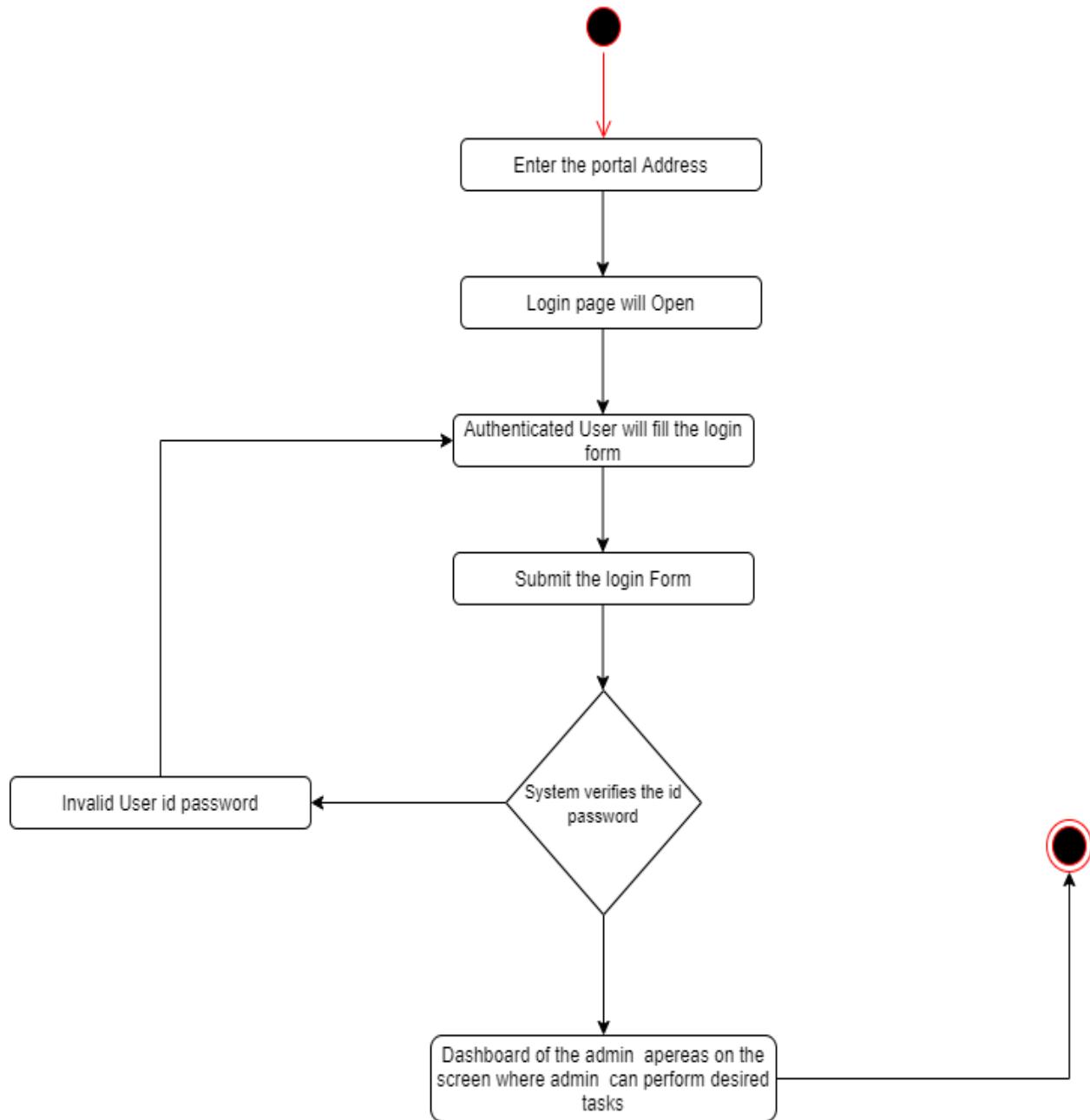


Figure 65 Activity Diagram for Login

4.5.2 Forget ID or Password

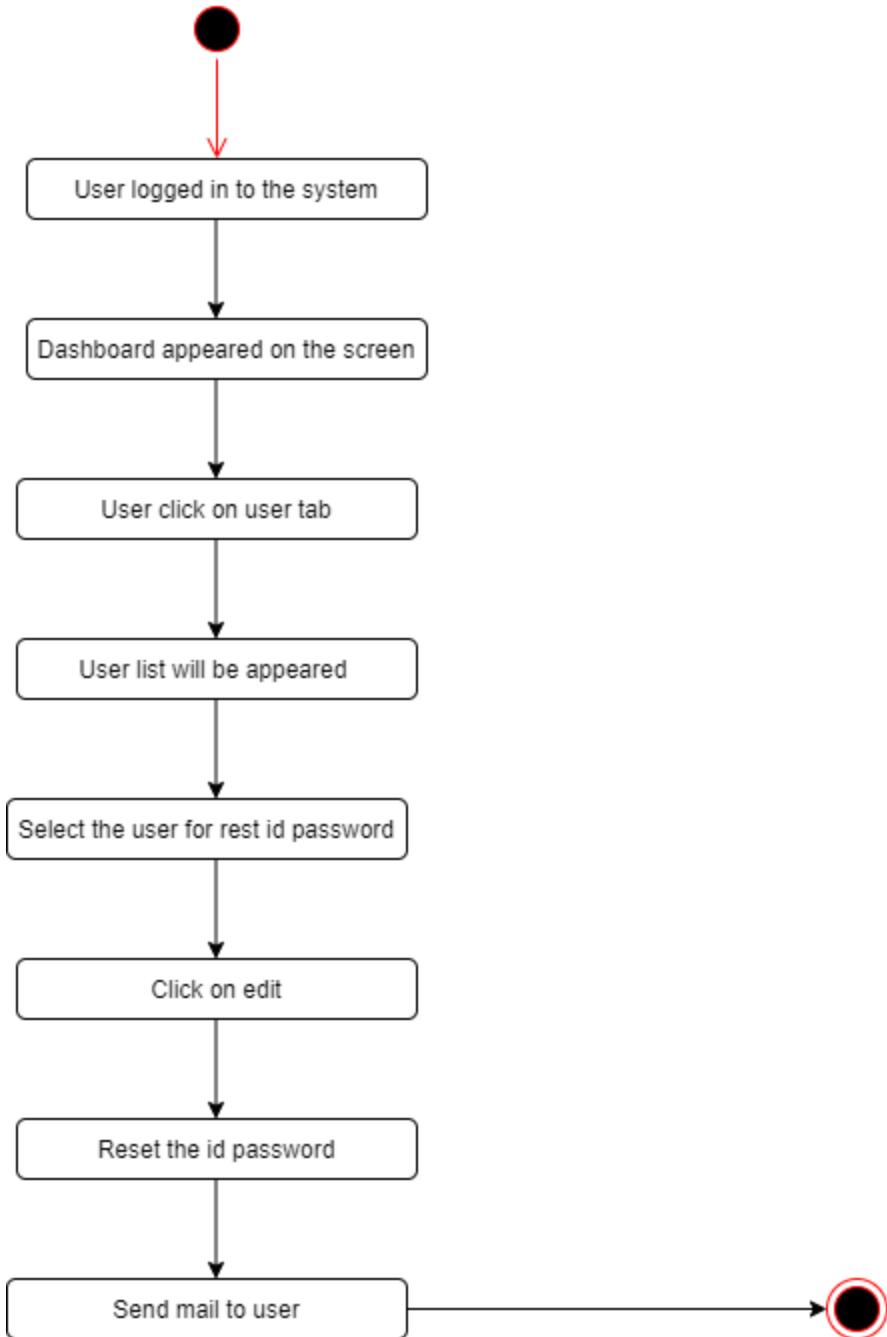


Figure 66 Activity Diagram for Forget ID or Password

4.5.3 Register New User

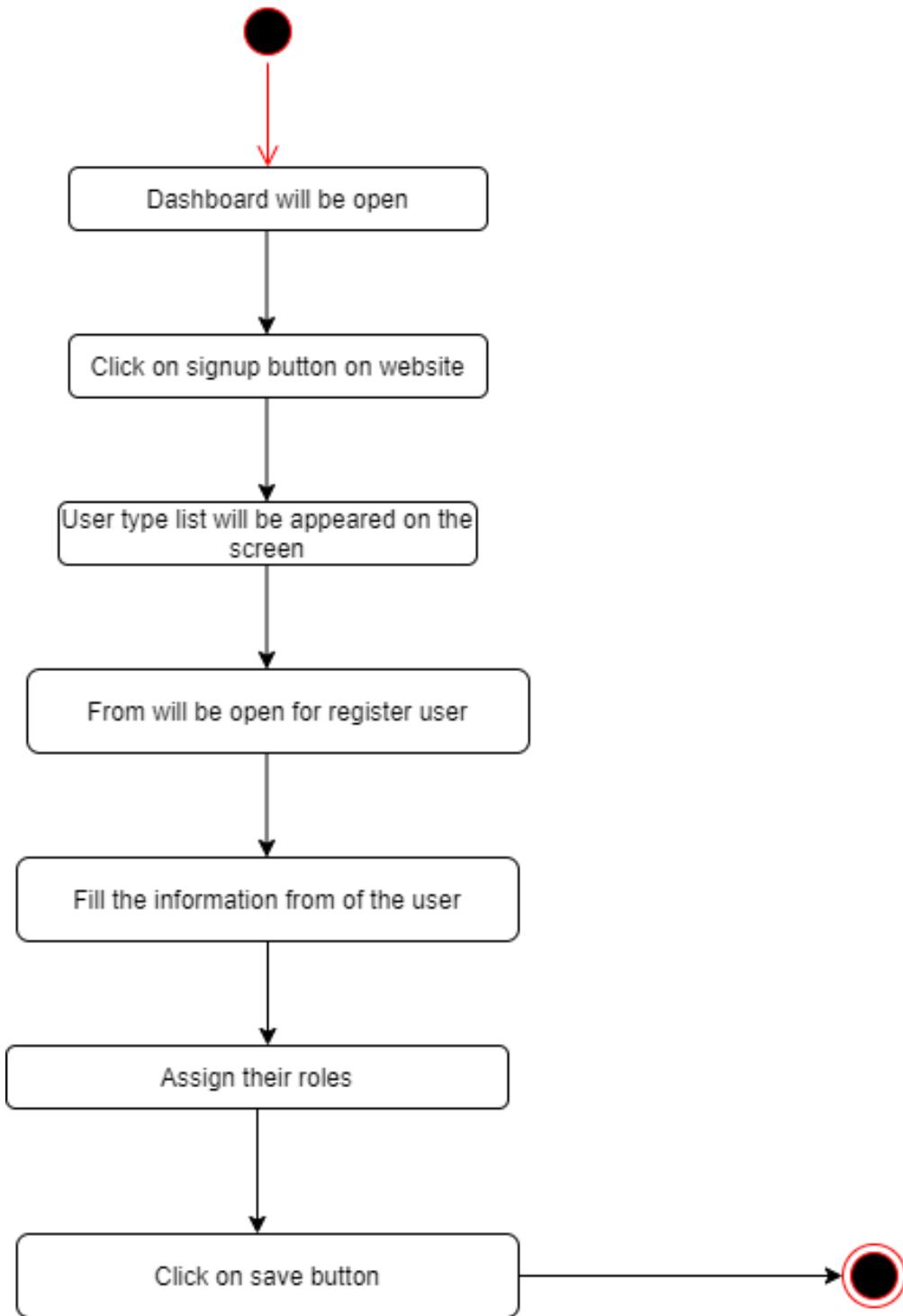


Figure 67 Activity Diagram for Register New User

4.5.4 Approve User Registration

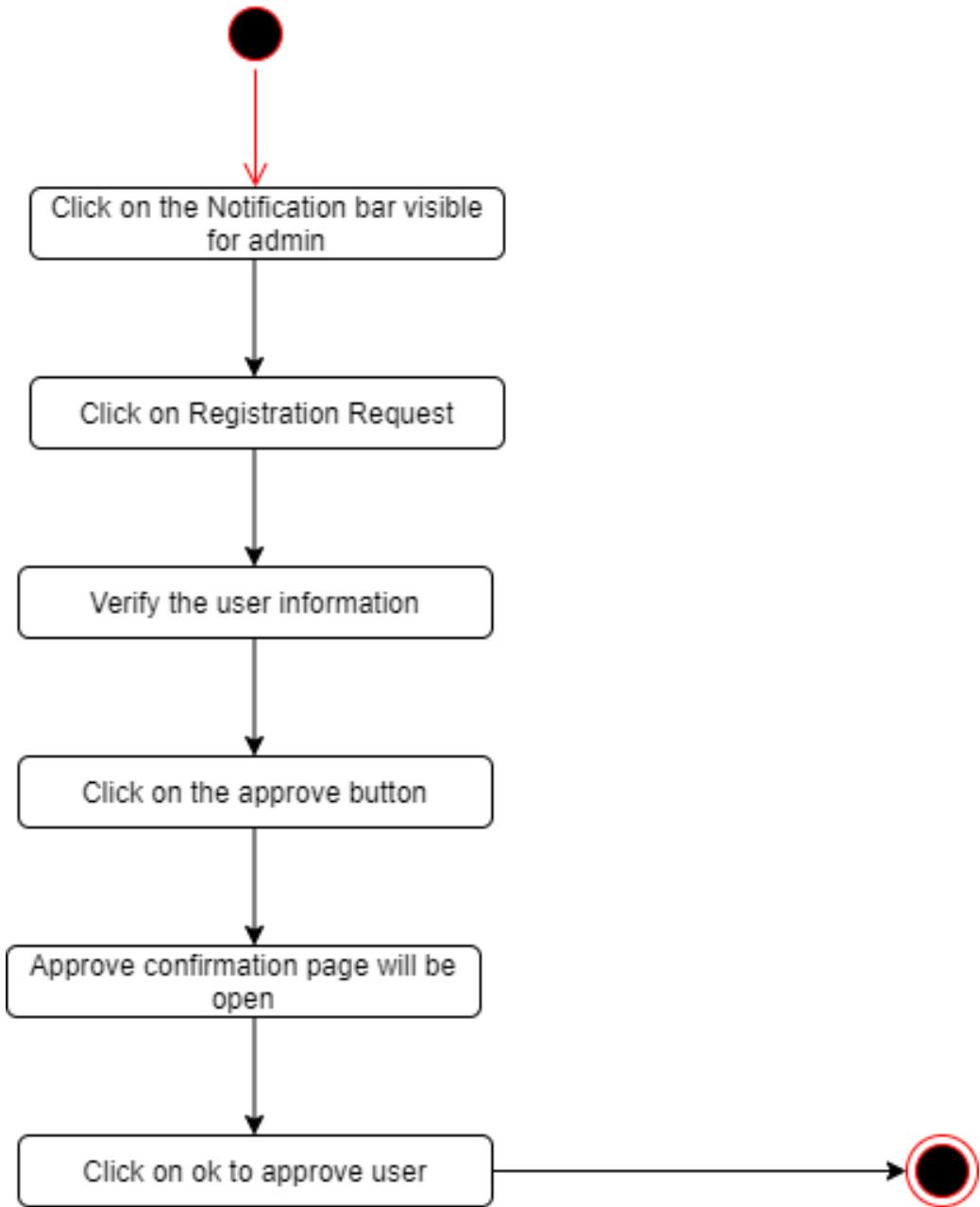


Figure 68 Activity Diagram for Approve User Registration

4.5.5 Block User

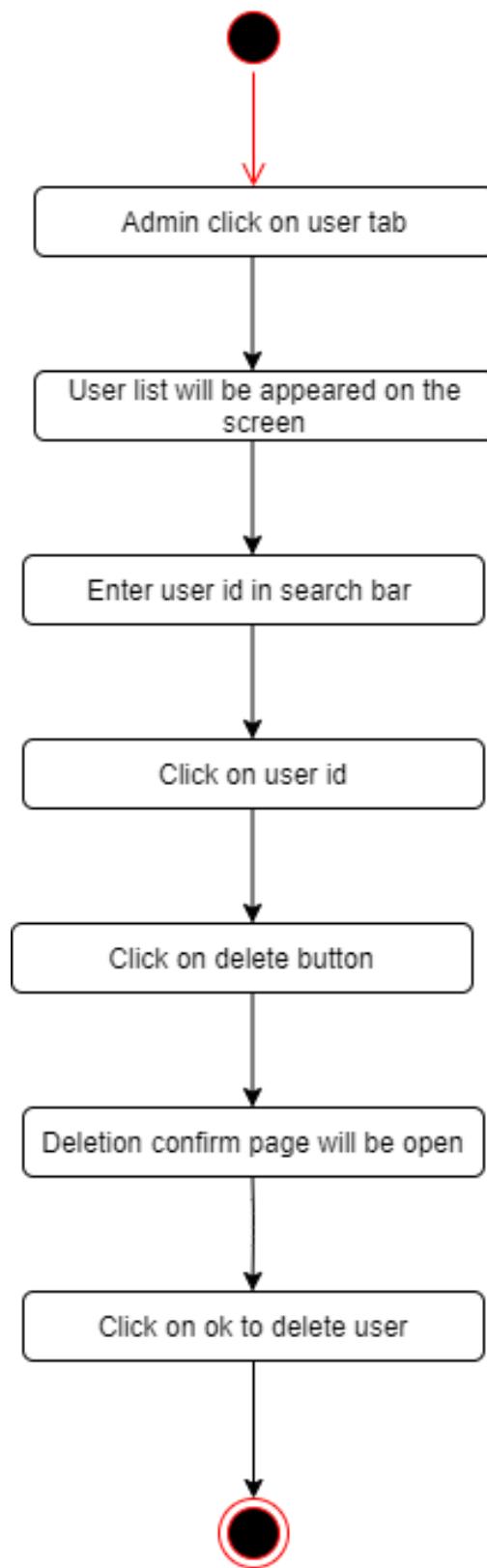


Figure 69 Activity Diagram for Block User

4.5.6 Assign Specific Role

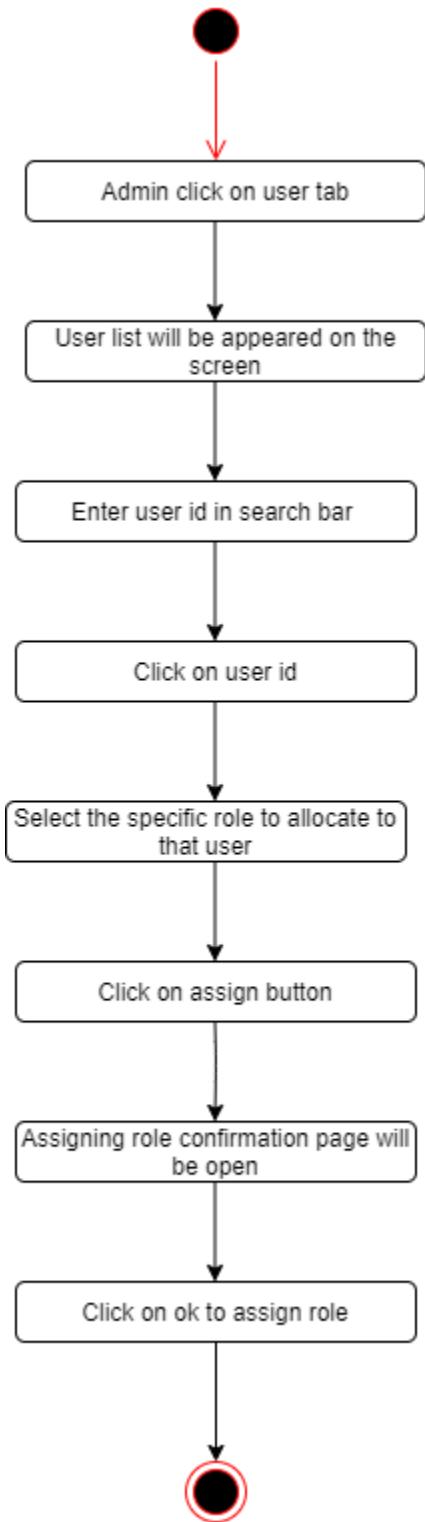


Figure 70 Activity Diagram for Assign Specific Role

4.5.7 Approve User's Queries

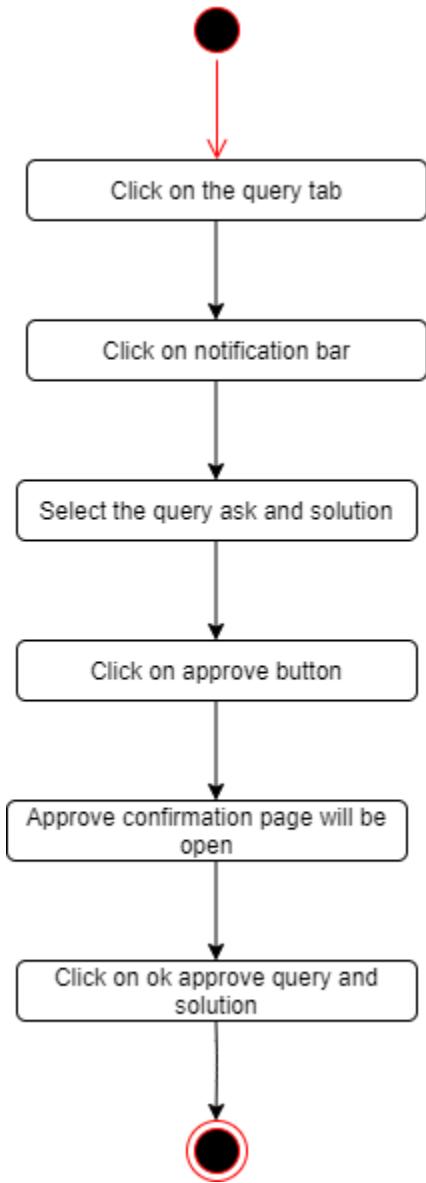


Figure 71 Activity Diagram for Approve User's Queries

4.5.8 Remove User's Queries

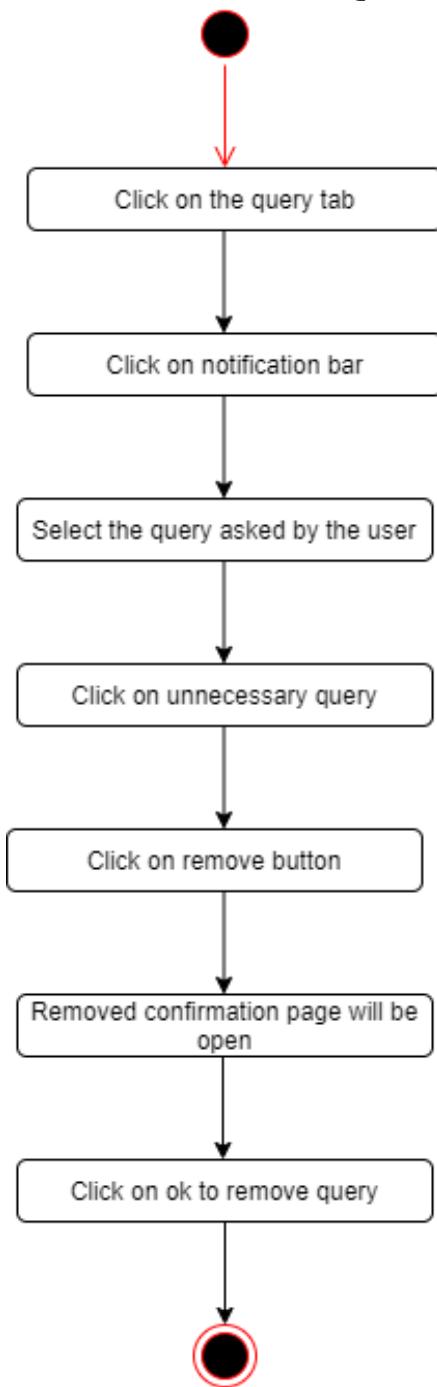


Figure 72 Activity Diagram for Remove User's Queries

4.5.9 Approve Developer' Script

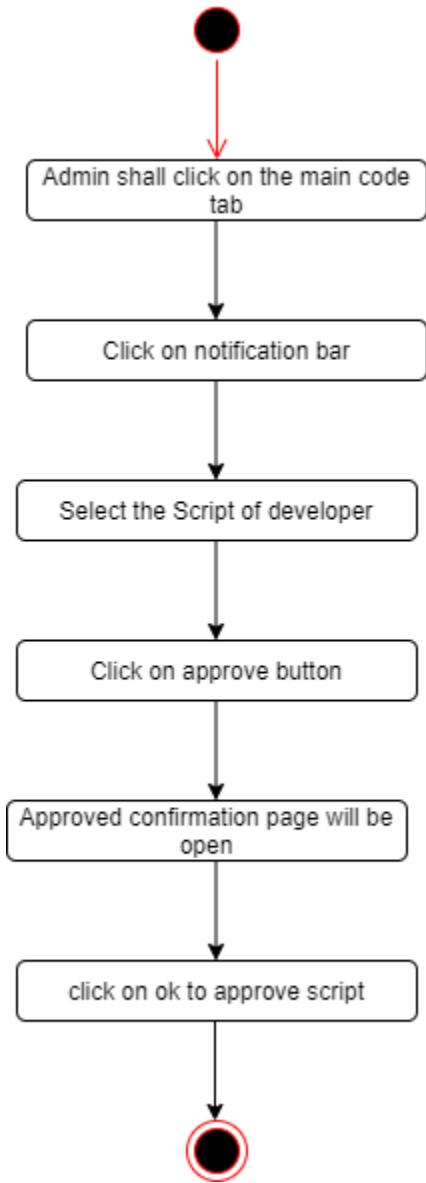


Figure 73 Activity Diagram for Approve Developer' Script

4.5.10 Write Code on Live Editor

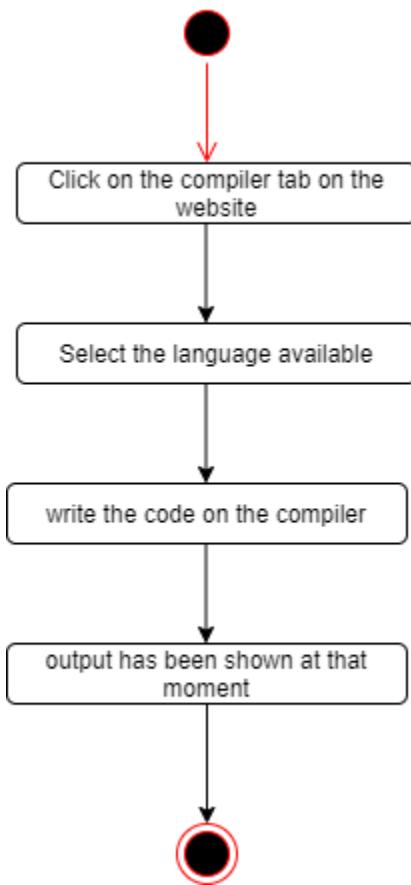


Figure 74 Activity Diagram for Write Code on Live Editor

4.5.11 save Code on Local Machine

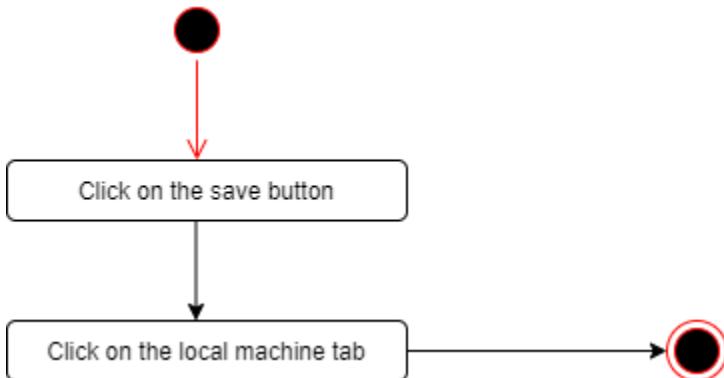


Figure 75 Activity Diagram for save Code on Local Machine

4.5.12 Allow Registered user to Save Code in Repository

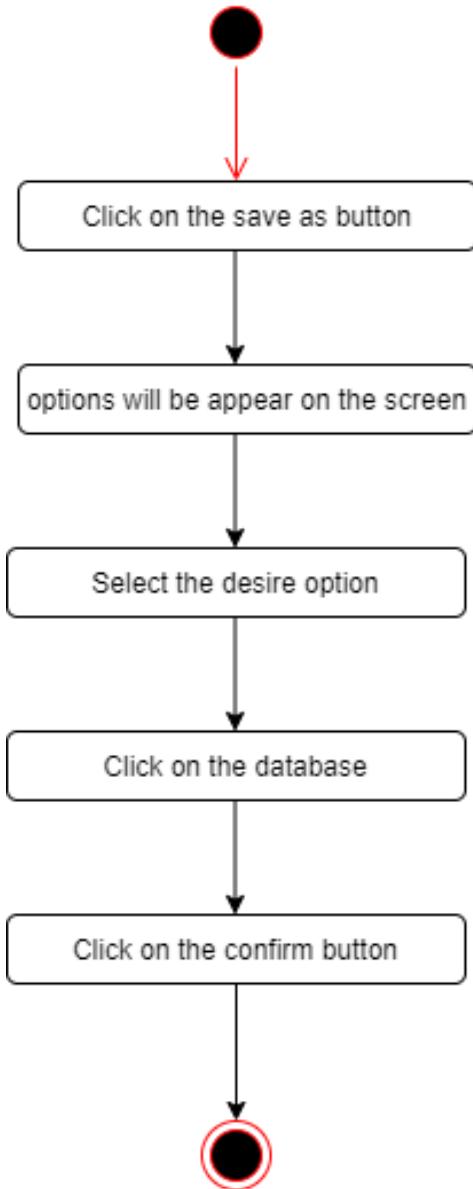


Figure 76 Activity Diagram for Allow Registered user to Save Code in Repository

4.5.13 Edit User Code

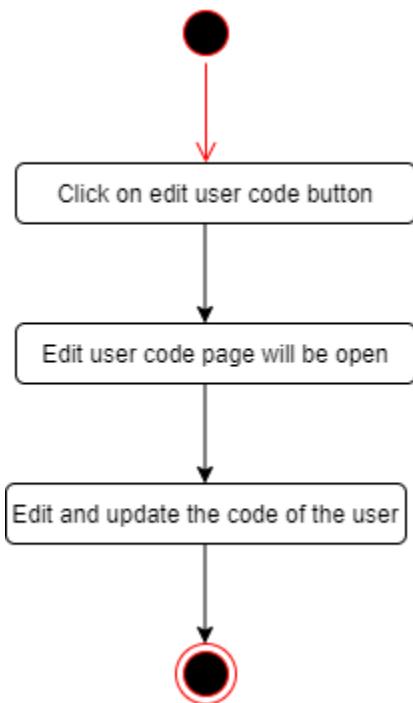


Figure 77 Activity Diagram for Edit User Code

4.5.14 View User Code Details

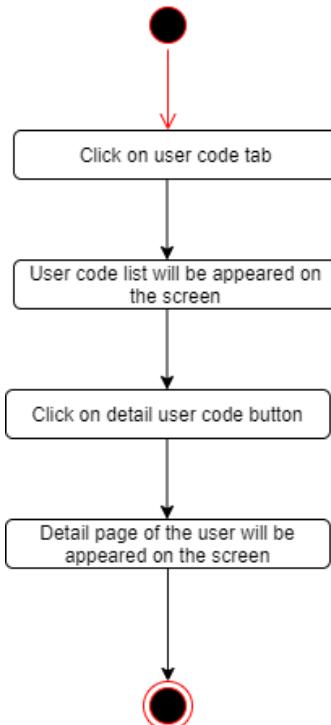


Figure 78 Activity Diagram for View User Code Details

4.5.15 Add Code to Database

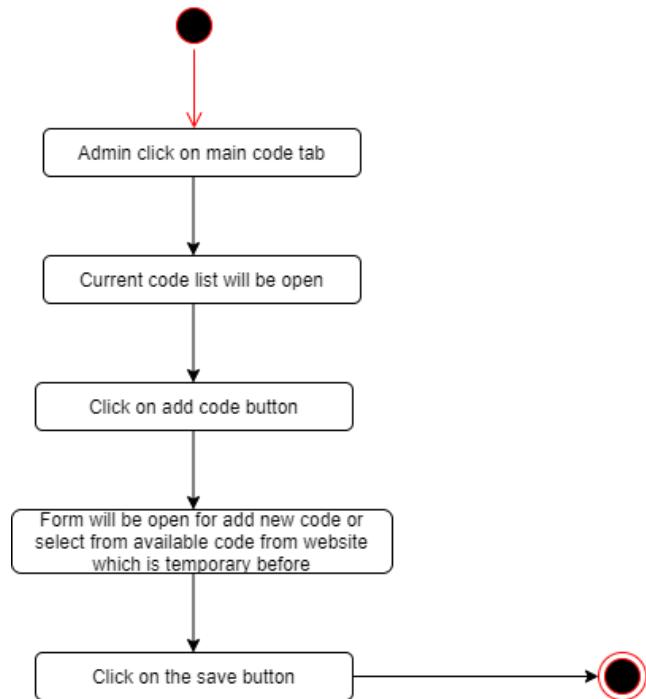


Figure 79 Activity Diagram for Add Code to Database

4.5.16 View Code on Database

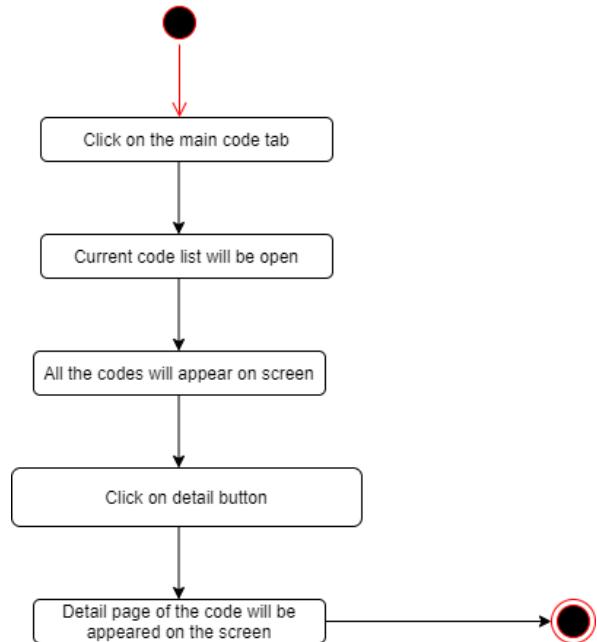


Figure 80 Activity Diagram for View Code on Database

4.5.17 Delete Code

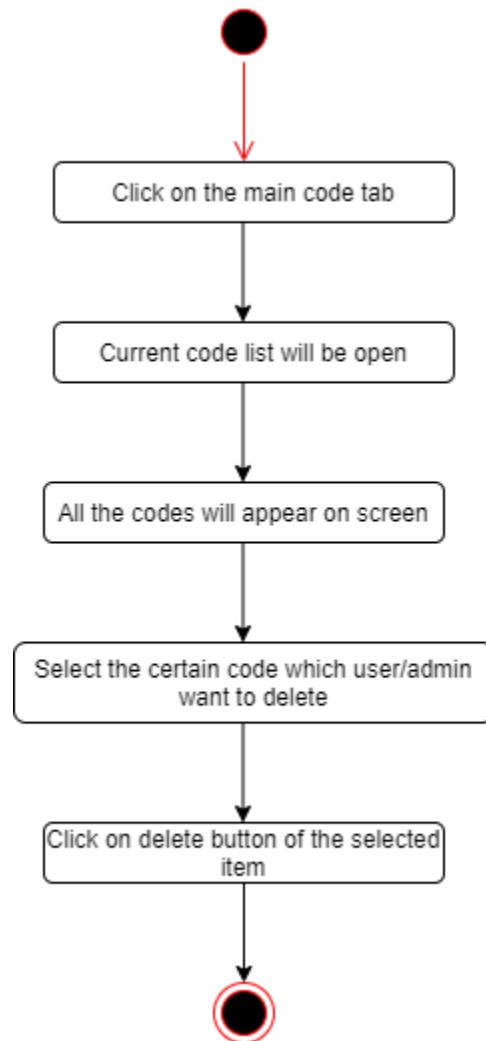


Figure 81 Activity Diagram for Delete Code

4.5.18 Payment Gateway Integration

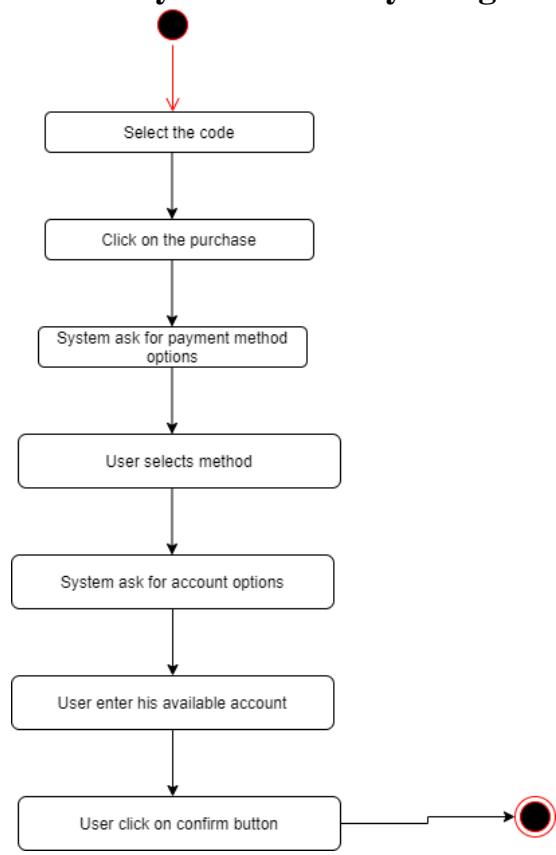


Figure 82 Activity Diagram for Payment Gateway Integration

4.5.19 Sales and Purchase

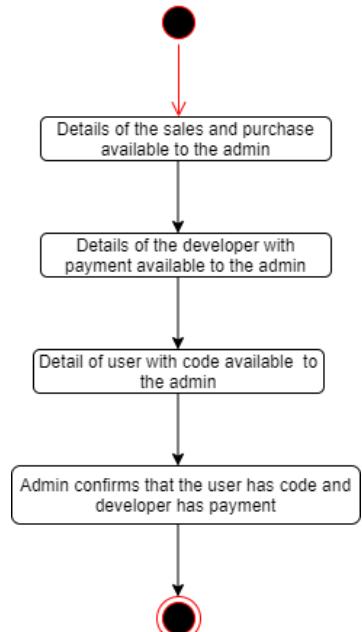


Figure 83 Activity Diagram for Sales and Purchase

4.5.20 Multiple Currency

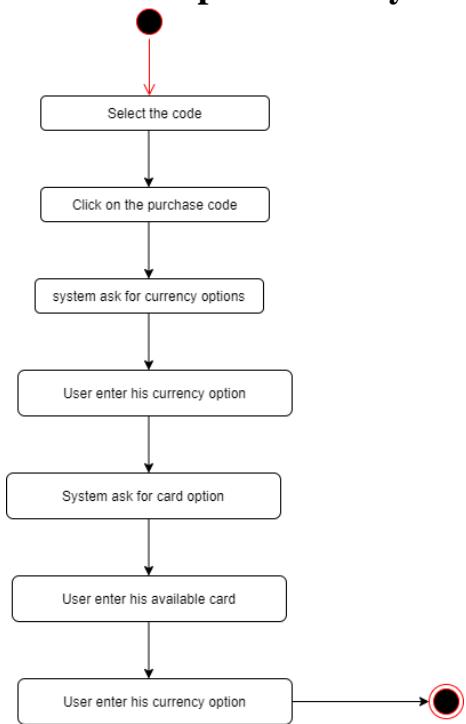


Figure 84 Activity Diagram for Multiple Currency

4.5.21 Developer Sales Code

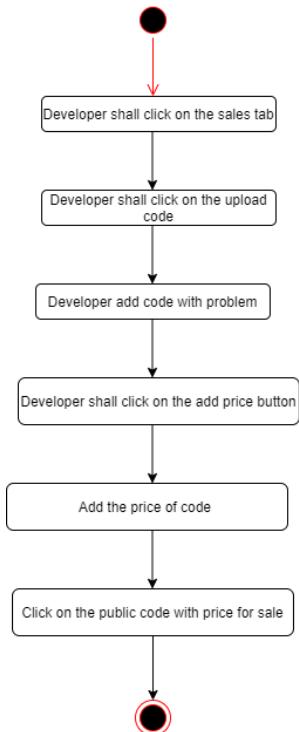


Figure 85 Activity Diagram for Developer Sales Code

4.5.22 User Purchase Code

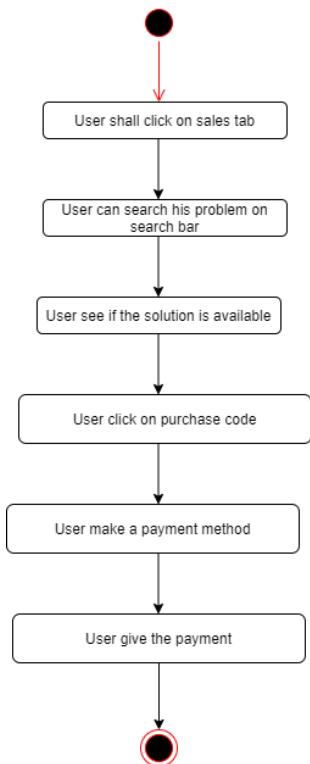


Figure 86 Activity Diagram for User Purchase Code

4.5.23 Payment Report

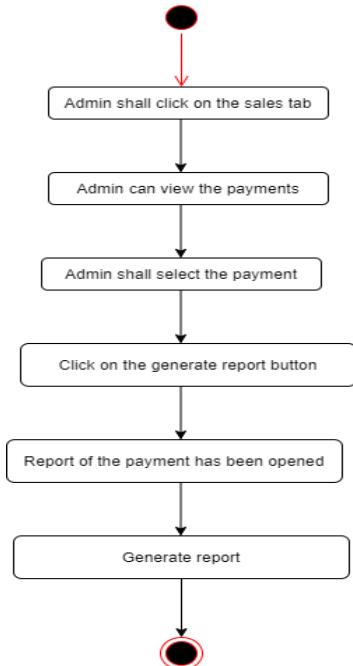


Figure 87 Activity Diagram for Payment Report

4.5.24 Review on Codes

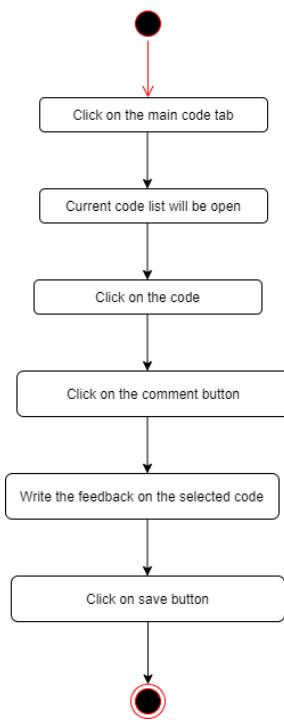


Figure 88 Activity Diagram for Review on Codes

4.5.25 Ratings

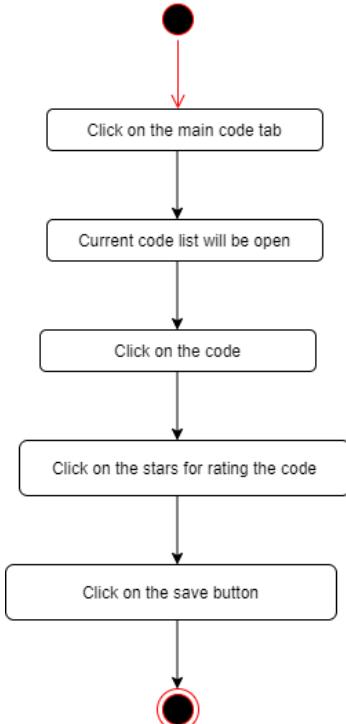


Figure 89 Activity Diagram for Ratings

4.5.26 Ask Query

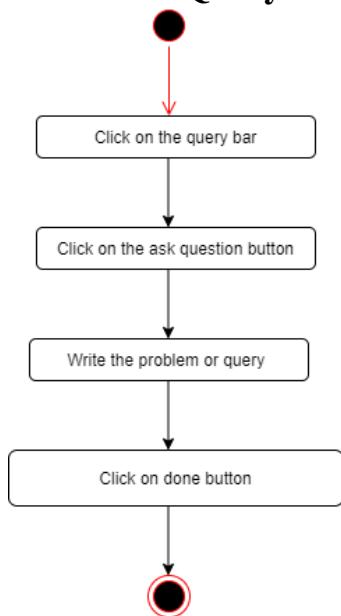


Figure 90 Activity Diagram for Ask Query

4.5.27 Solution of Query

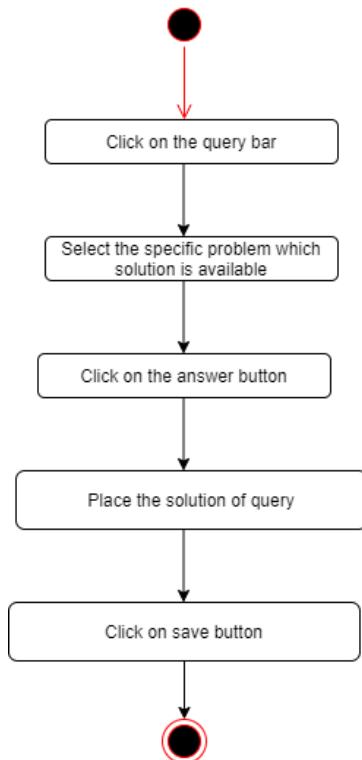


Figure 91 Activity Diagram for Solution of Query

4.5.28 Admin Upload Codes

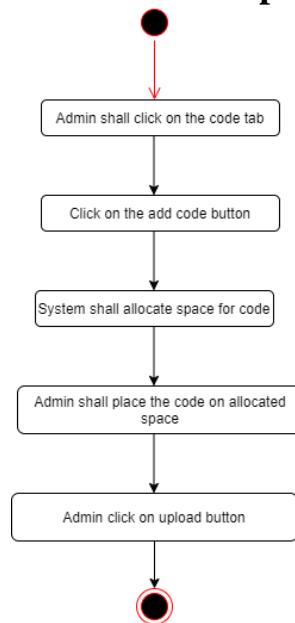


Figure 92 Activity Diagram for Admin Upload Codes

4.5.29 User Upload Codes

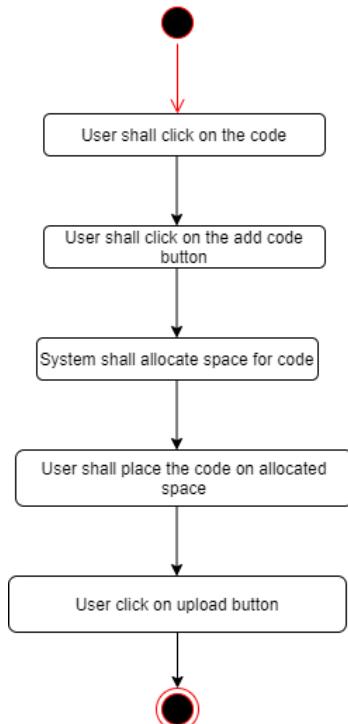


Figure 93 Activity Diagram for User Upload Codes

4.5.30 User Rate Script

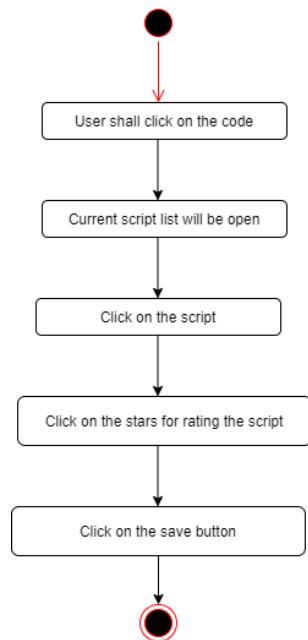


Figure 94 Activity Diagram for User Rate Script

4.5.31 Allow User to Switch Role

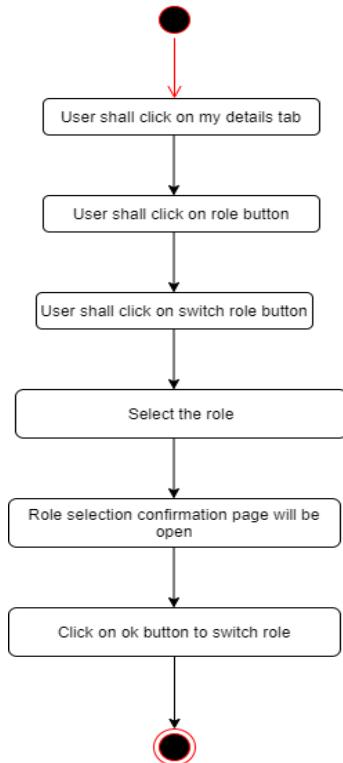


Figure 95 Activity Diagram for Allow User to Switch Role

4.5.32 User Participate In Voting

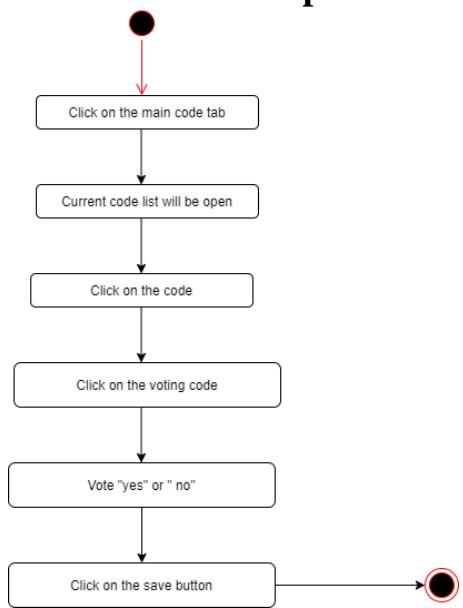


Figure 96 Activity Diagram for User Participate In Voting

4.5.33 Upload Script

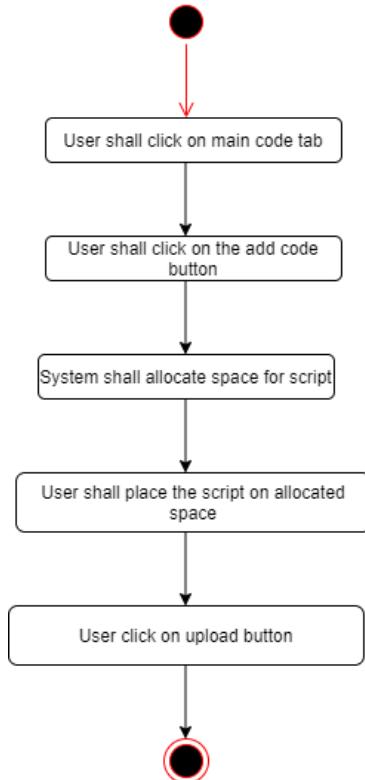


Figure 97 Activity Diagram for Upload Script

4.5.34 Delete Script

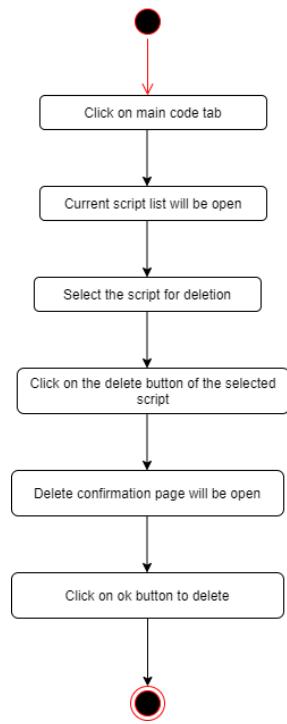


Figure 98 Activity Diagram for Delete Script

4.5.35 Update Script

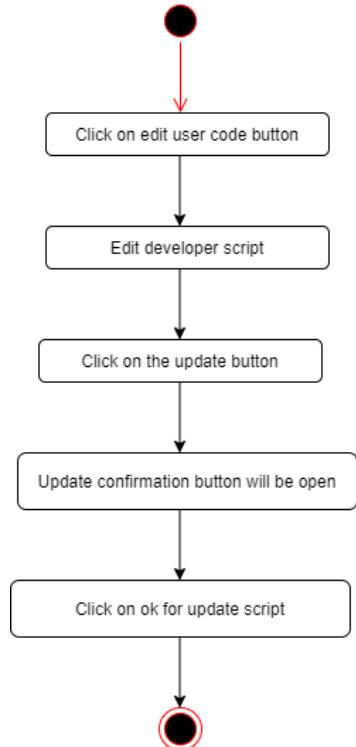


Figure 99 Activity Diagram for Update Script

4.5.36 User Ask Query

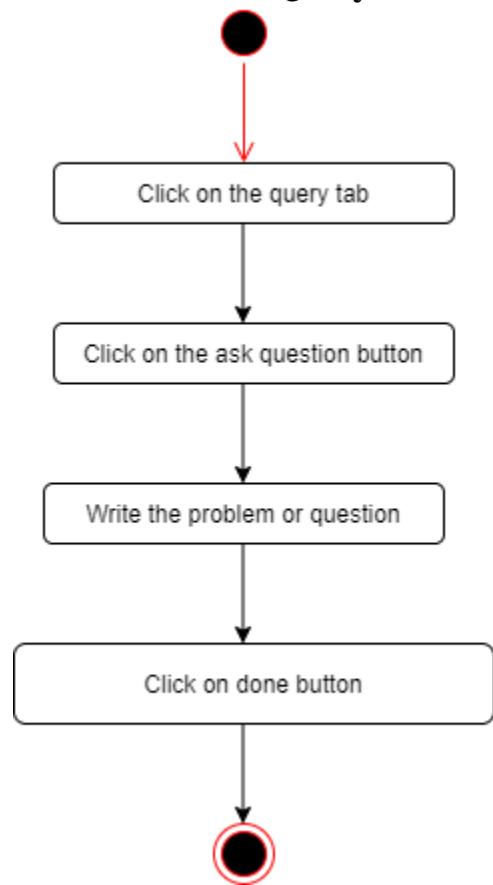


Figure 100 Activity Diagram for User Ask Query

User Panel

4.5.37 Login

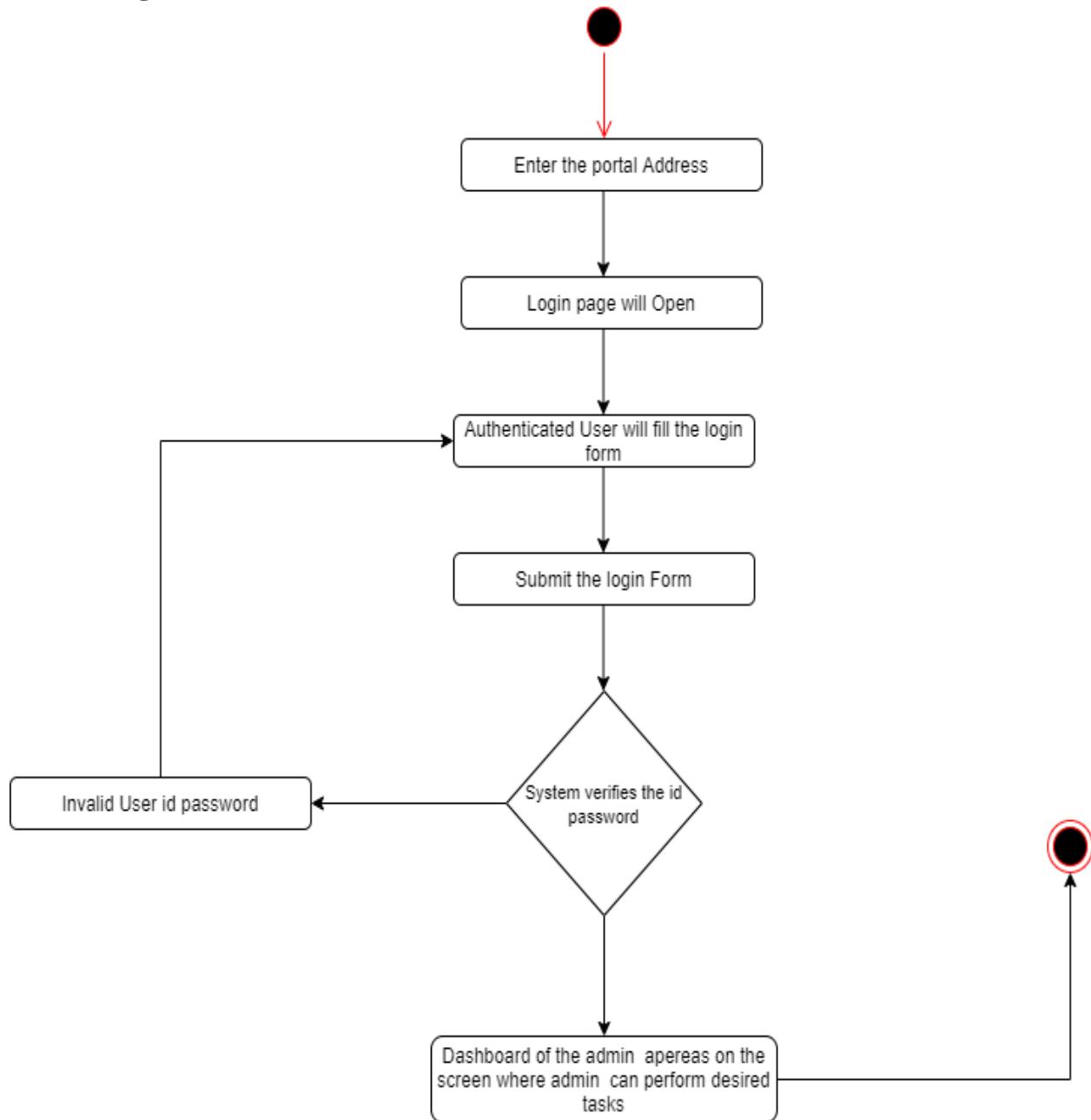


Figure 101 Activity Diagram for Login

4.5.38 Forget ID or Password

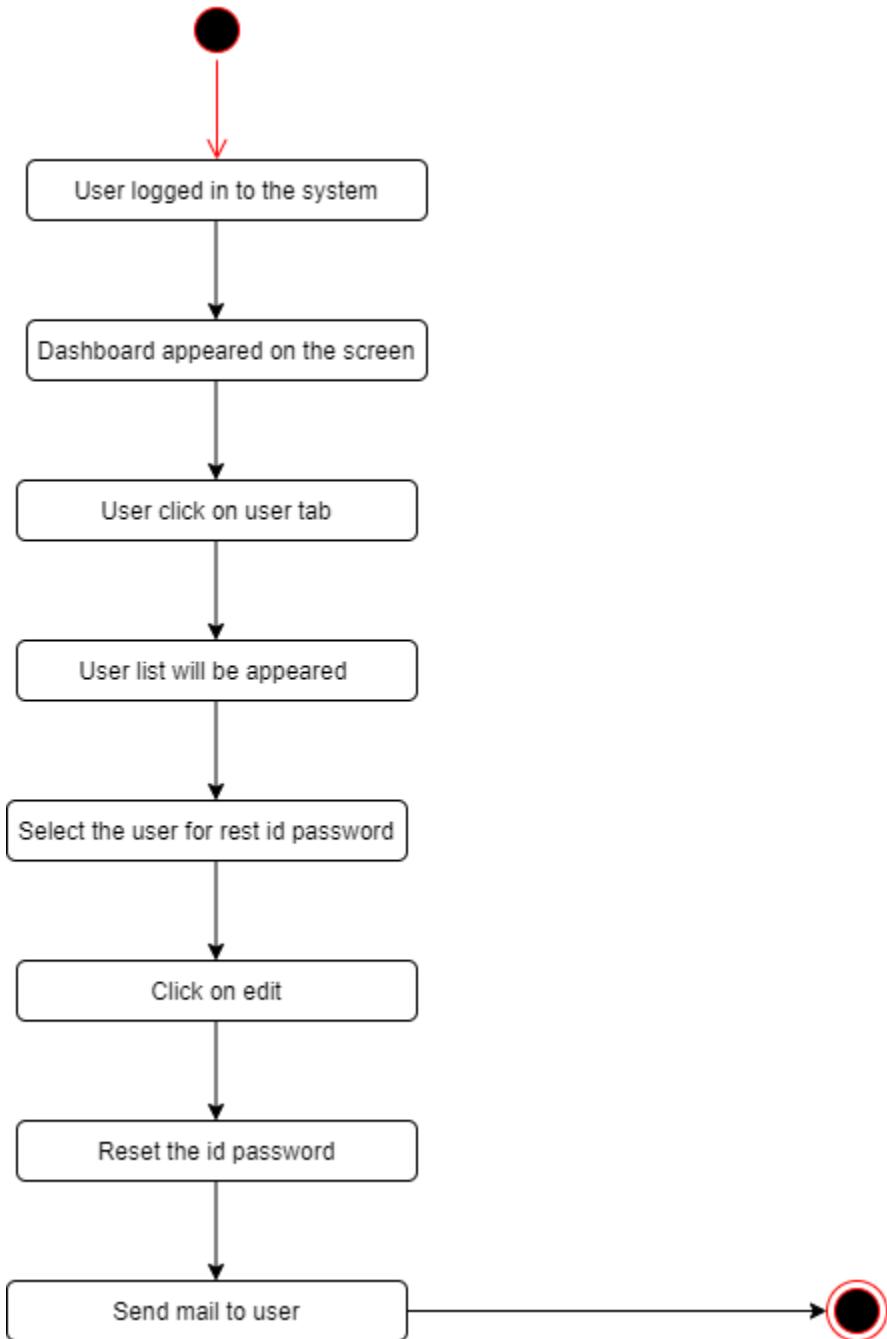


Figure 102 Activity Diagram for Forget ID or Password

4.5.39 Write Code on Live Editor

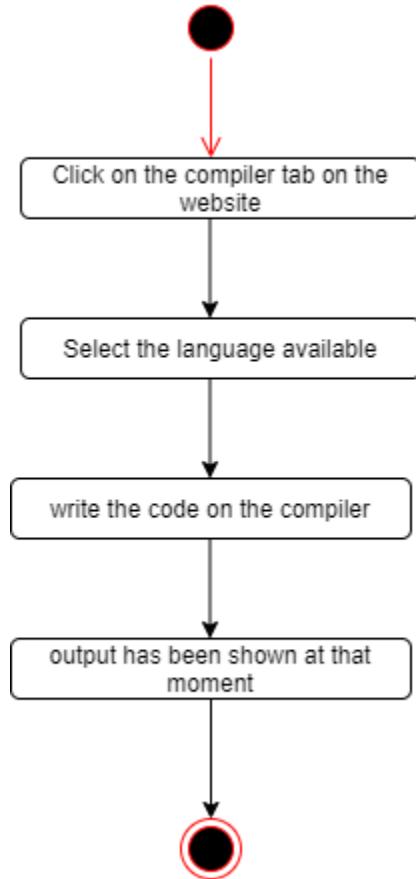


Figure 103 Activity Diagram for Write Code on Live Editor

4.5.40 save Code on Local Machine

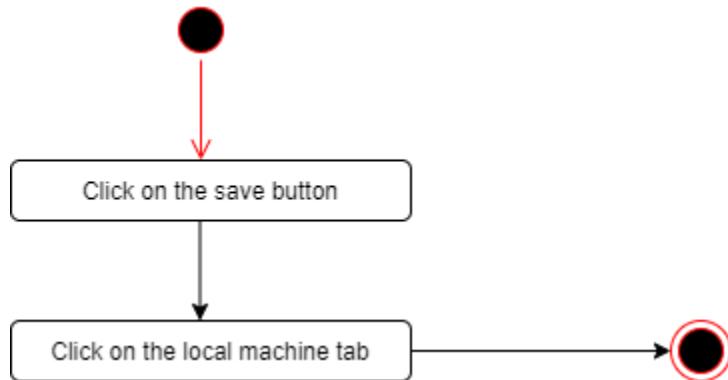


Figure 104 Activity Diagram for save Code on Local Machine

4.5.41 Allow Registered User to Save Code In Repository

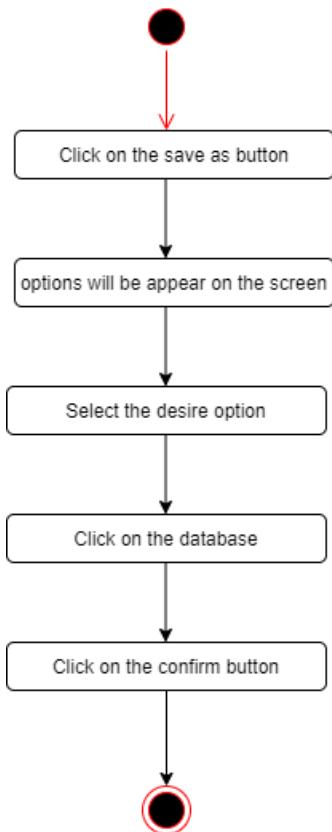


Figure 105 Activity Diagram for Allow Registered User to Save Code In Repository

4.5.42 Sales and Purchase

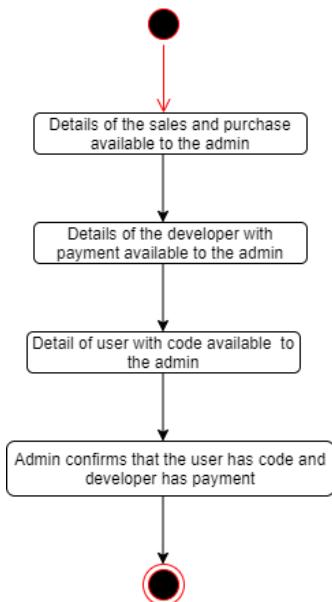


Figure 106 Activity Diagram for Sales and Purchase

4.5.43 Multiple Currency

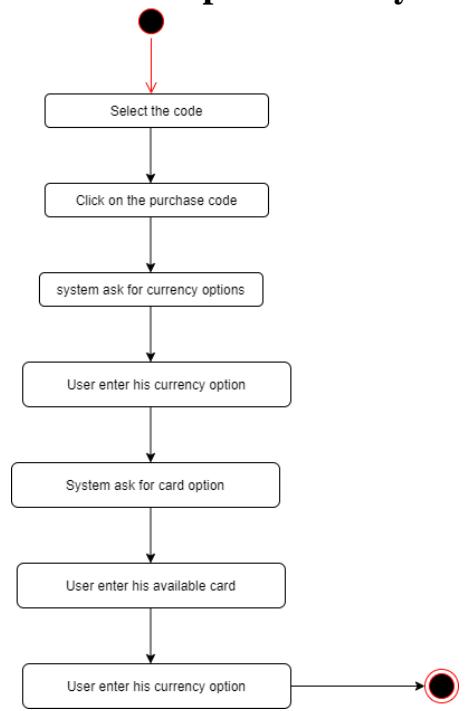


Figure 107 Activity Diagram for Multiple Currency

4.5.44 User Purchase Code

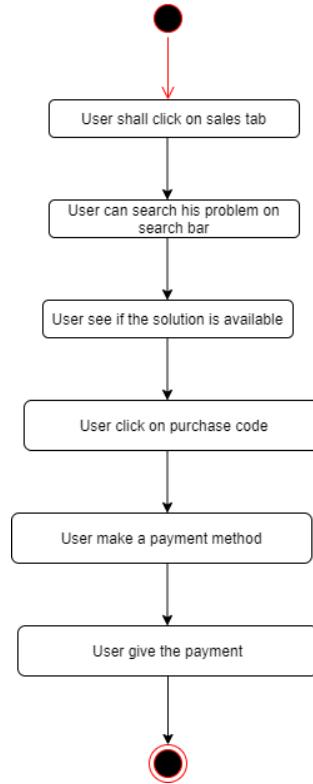


Figure 108 Activity Diagram for User Purchase Code

4.5.45 Payment Gateway Integration

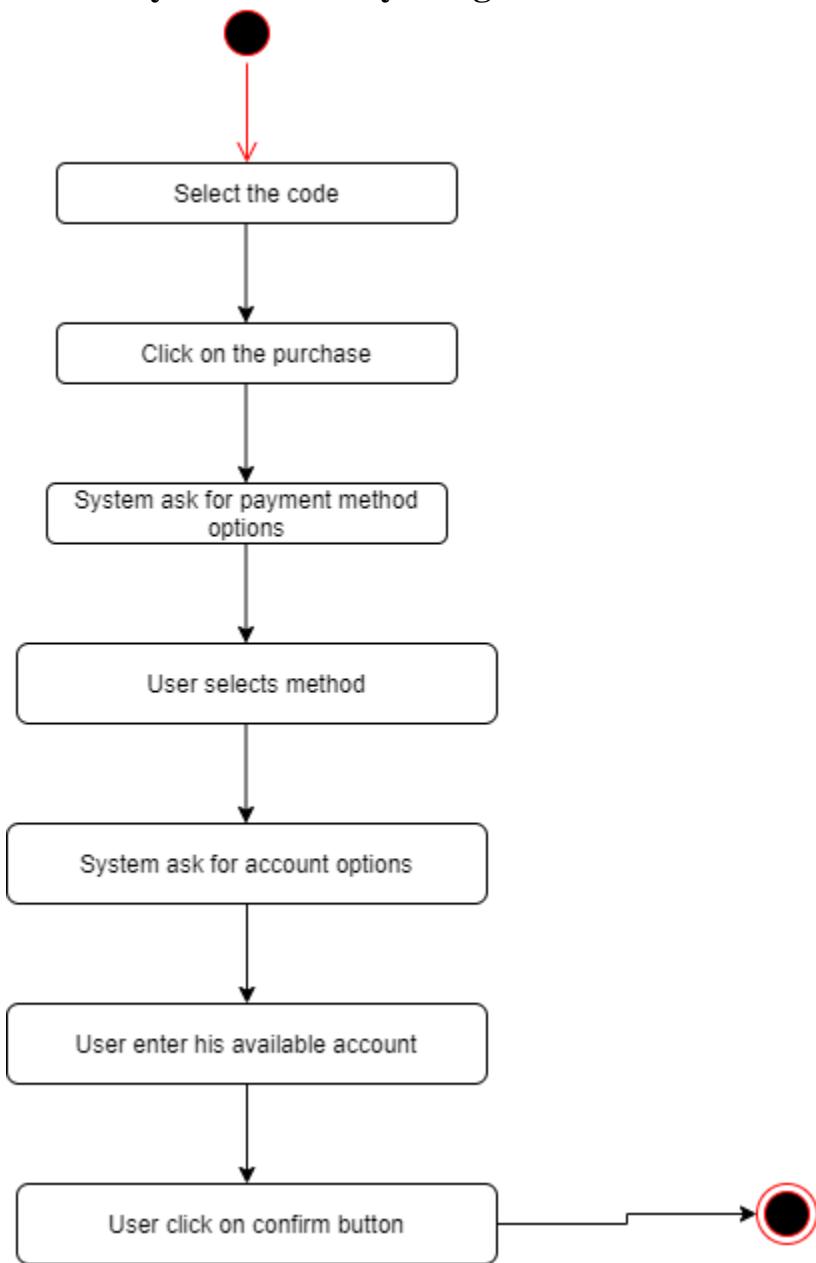


Figure 109 Activity Diagram for Payment Gateway Integration

4.5.46 Review on Codes

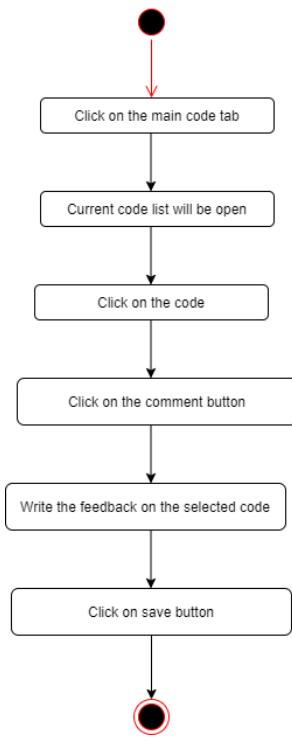


Figure 110 Activity Diagram for Review on Codes

4.5.47 Ratings

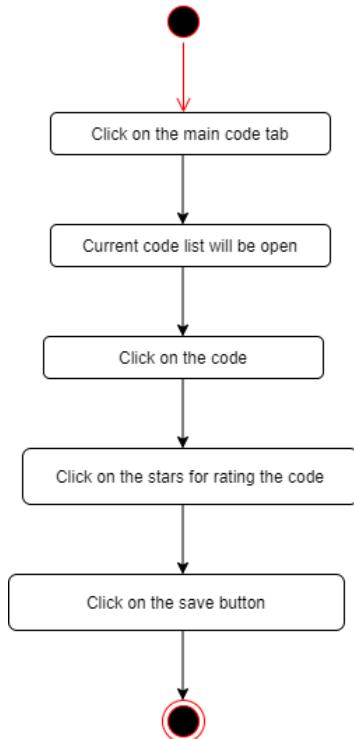


Figure 111 Activity Diagram for Ratings

4.5.48 Ask Query

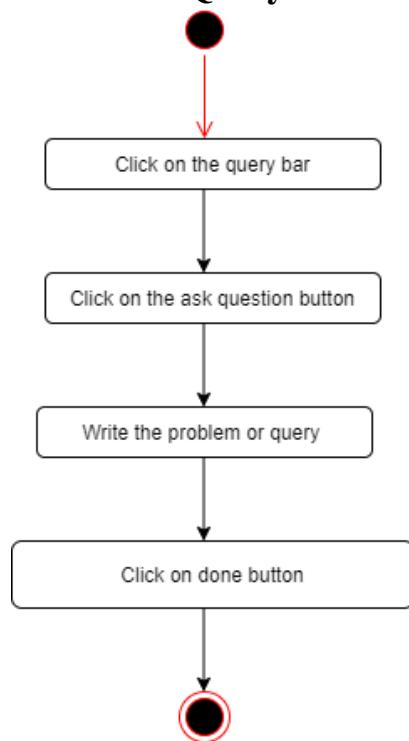


Figure 112 Activity Diagram for Ask Query

4.5.49 Solution of Query

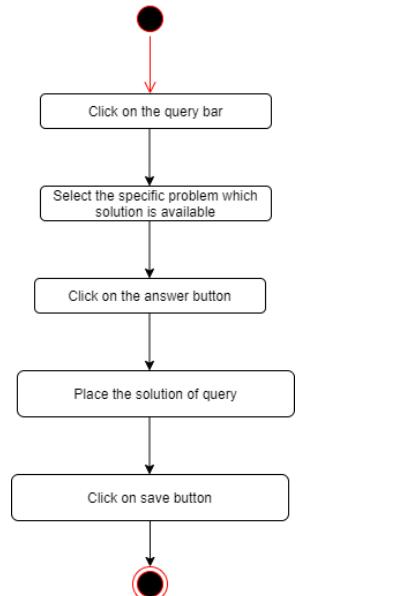


Figure 113 Activity Diagram for Solution of Query

4.5.50 User Upload Codes

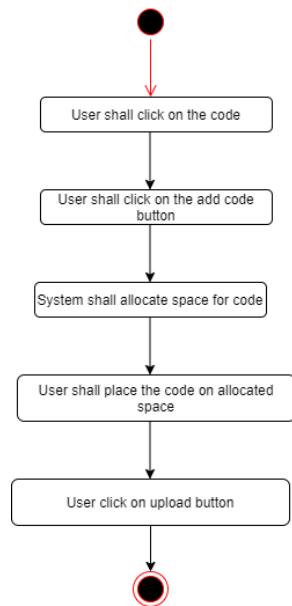


Figure 114 Activity Diagram for User Upload Codes

4.5.51 User Rate Script

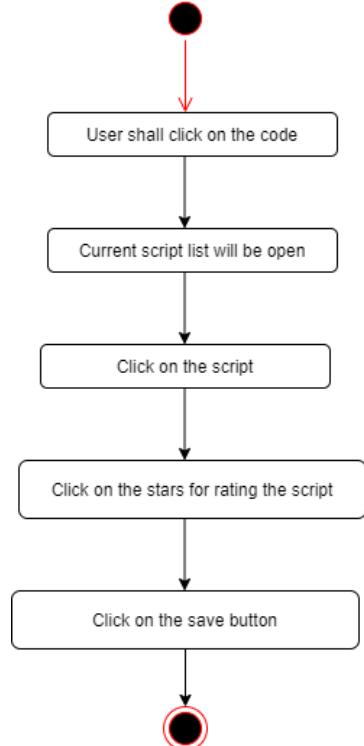


Figure 115 Activity Diagram for User Rate Script

4.5.52 Allow User to Switch Role

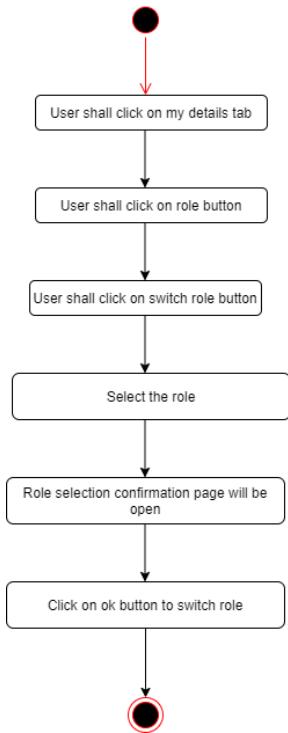


Figure 116 Activity Diagram for Allow User to Switch Role

4.5.53 User Participate In Voting

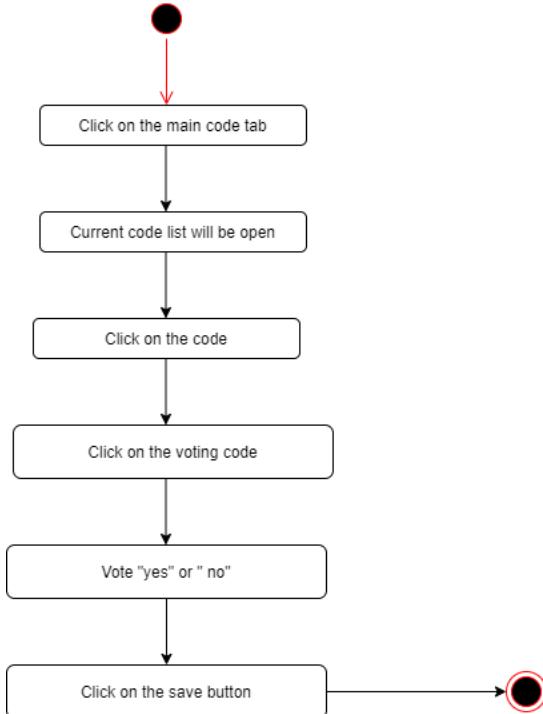


Figure 117 Activity Diagram for User Participate In Voting

4.5.54 Delete Script

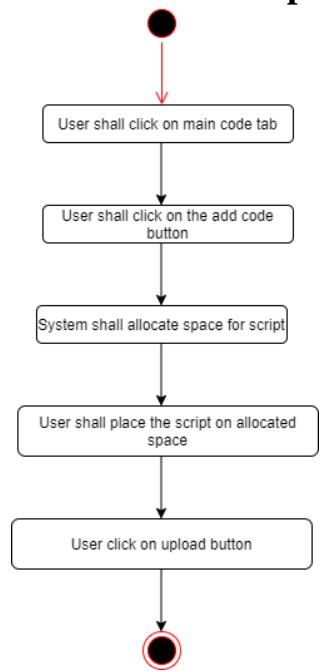


Figure 118 Activity Diagram for Delete Script

4.5.55 Update Script

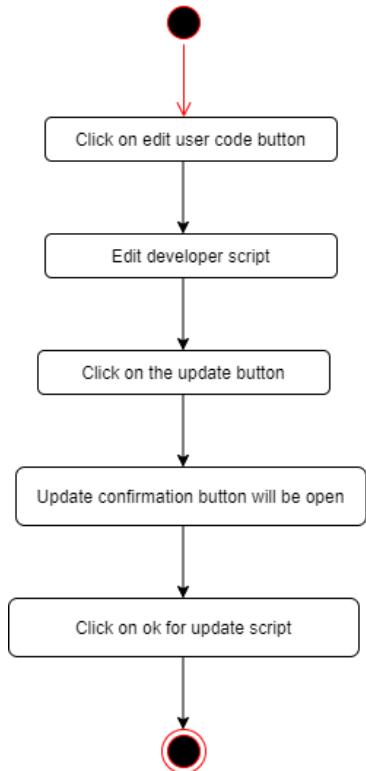


Figure 119 Activity Diagram for Update Script

4.6 Sequence Diagram

Admin Panel

4.6.1 Login

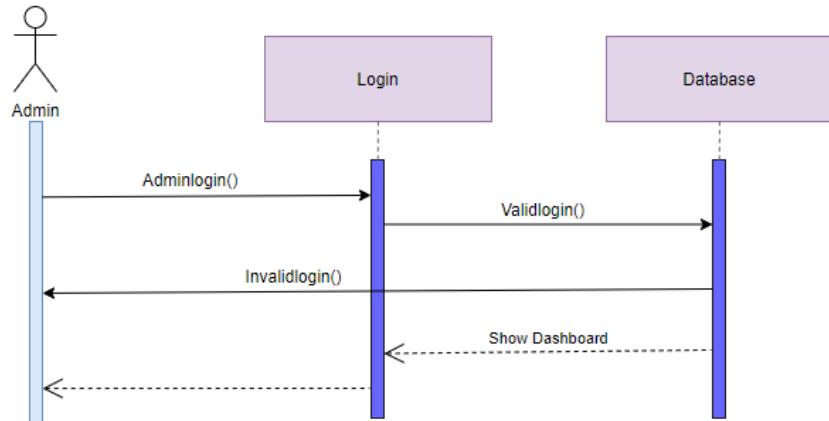


Figure 120 Sequence Diagram for Login

4.6.2 Forget ID or Password

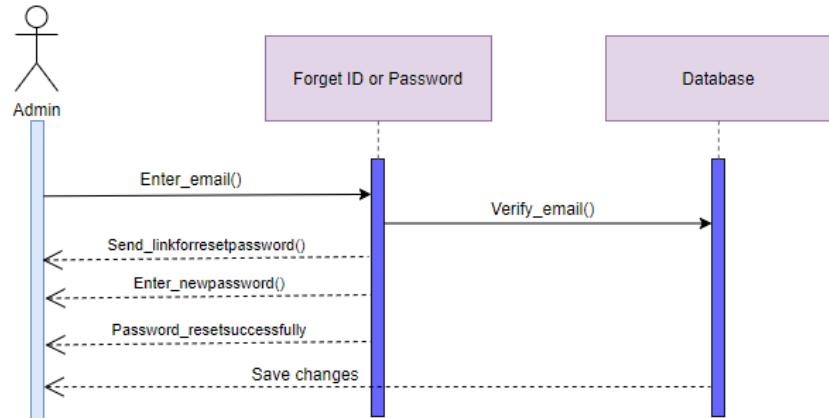


Figure 121 Sequence Diagram for Forget ID or Password

4.6.3 Register New User

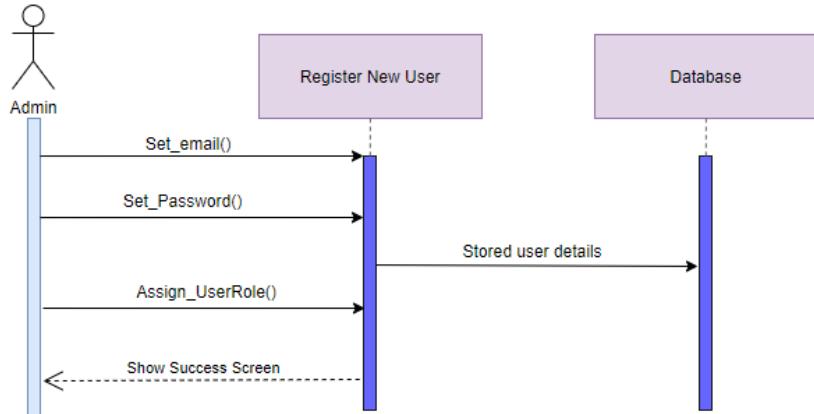


Figure 122 Sequence Diagram for Register New User

4.6.4 Approve User Registration

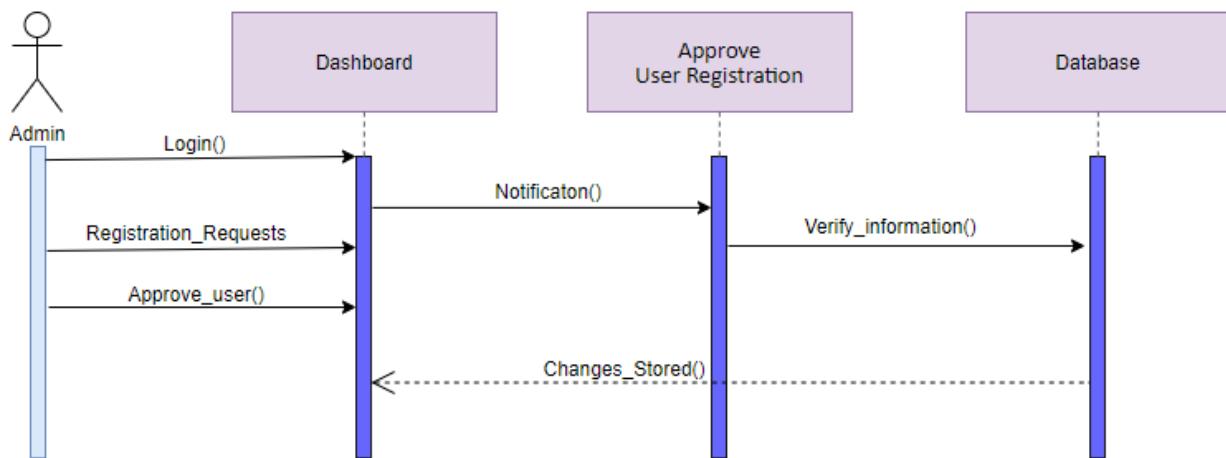


Figure 123 Sequence Diagram for Approve User Registration

4.6.5 Block User

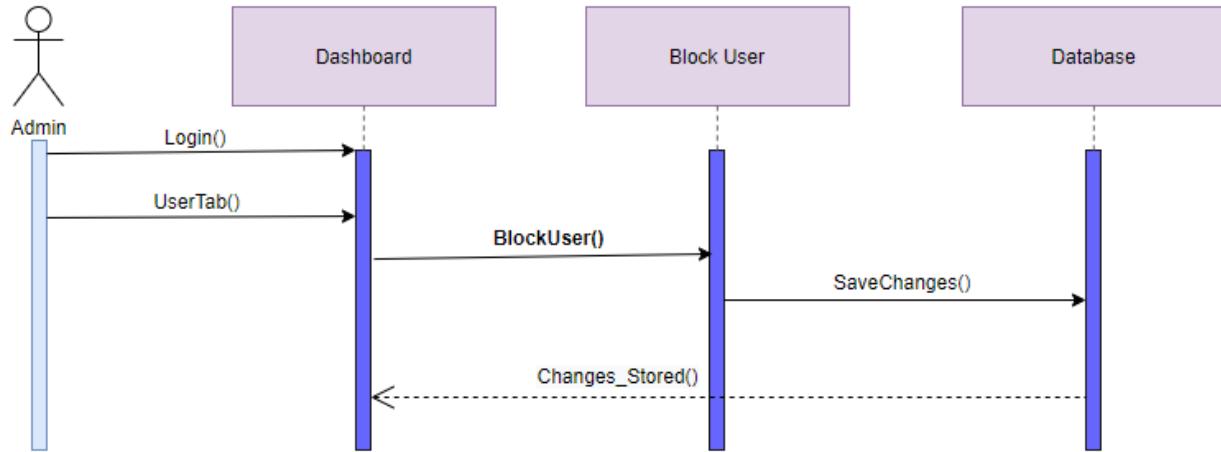


Figure 124 Sequence Diagram for Block User

4.6.6 Assign Specific Role

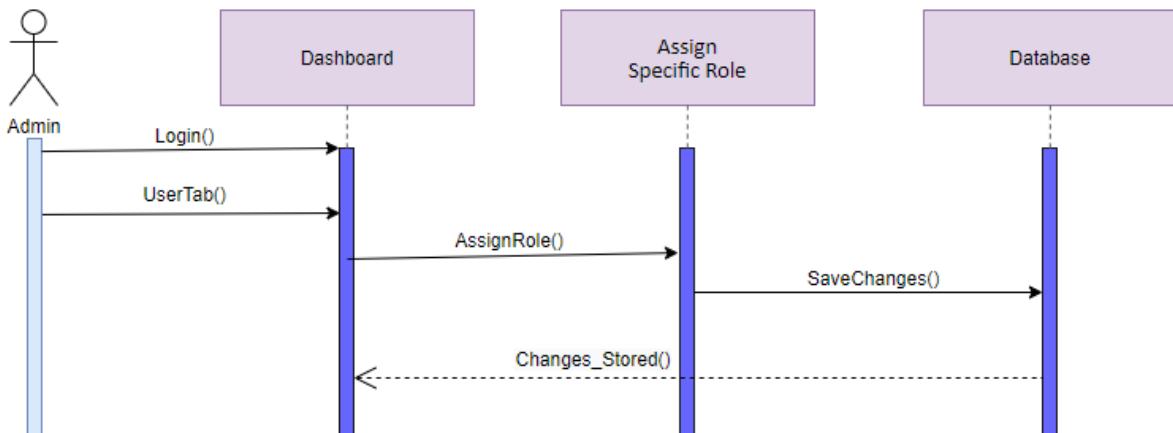


Figure 125 Sequence Diagram for Assign Specific Role

4.6.7 Approve User's Queries

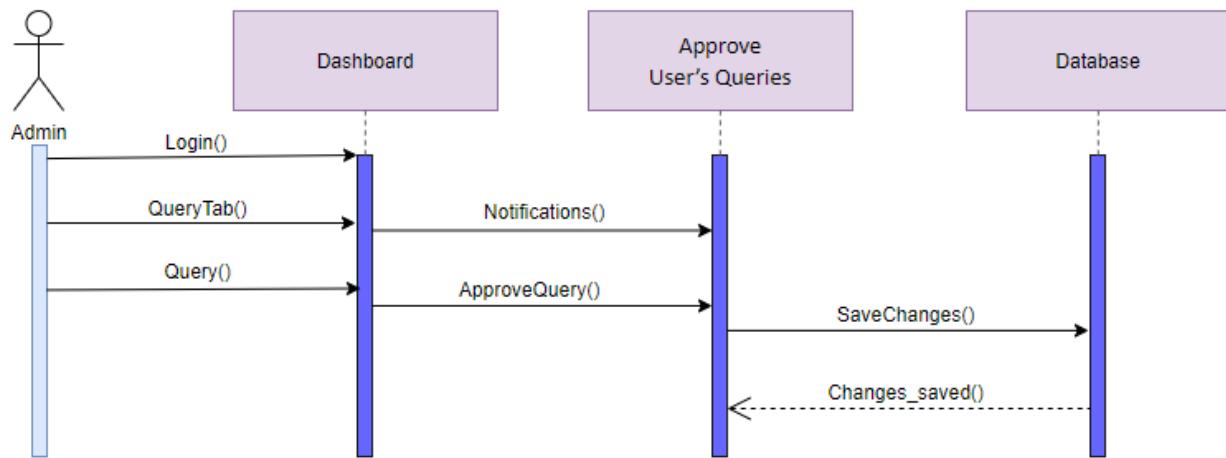


Figure 126 Sequence Diagram for Approve User's Queries

4.6.8 Remove User's Queries

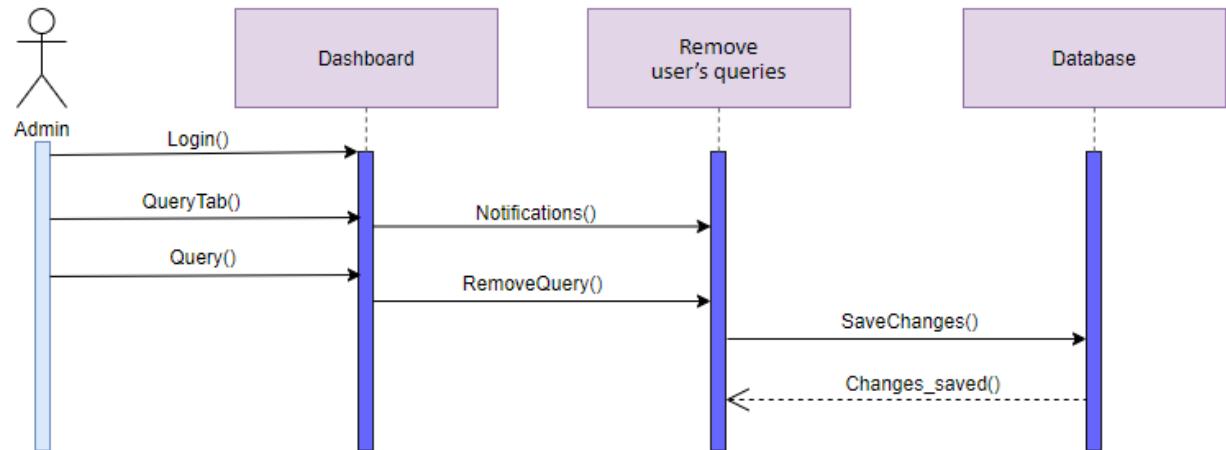


Figure 127 Sequence Diagram for Remove User's Queries

4.6.9 Approve Developer' Script

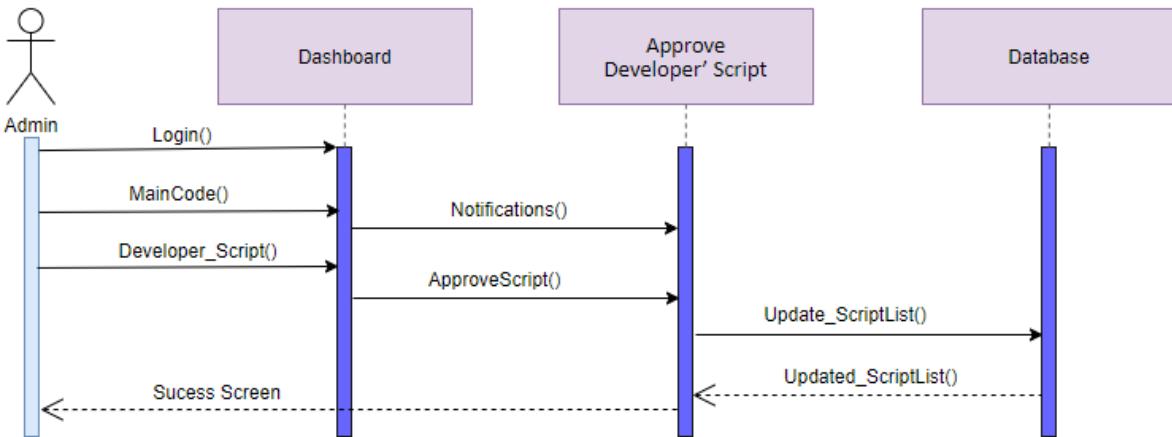


Figure 128 Sequence Diagram for Approve Developer' Script

4.6.10 Write Code on Live Editor

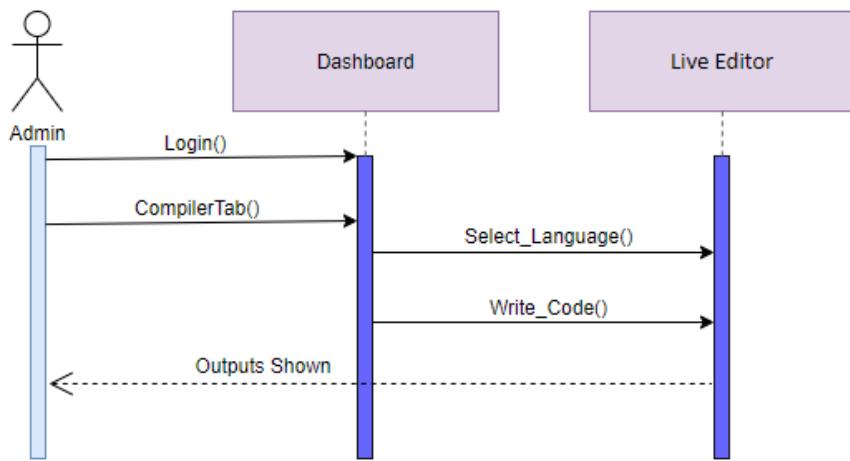


Figure 129 Sequence Diagram for Write Code on Live Editor

4.6.11 save Code on Local Machine

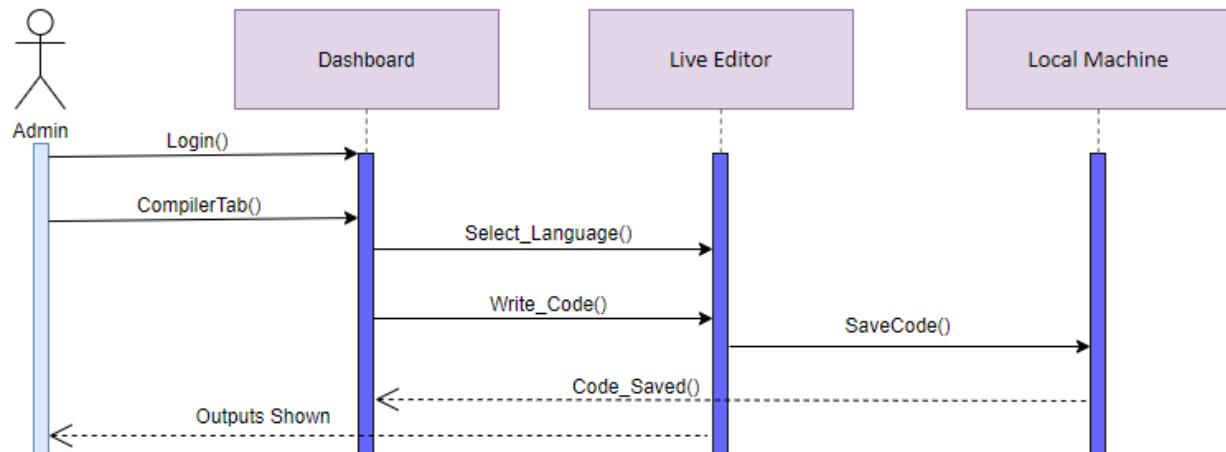


Figure 130 Sequence Diagram for save Code on Local Machine

4.6.12 Allow Registered user to Save Code in Repository

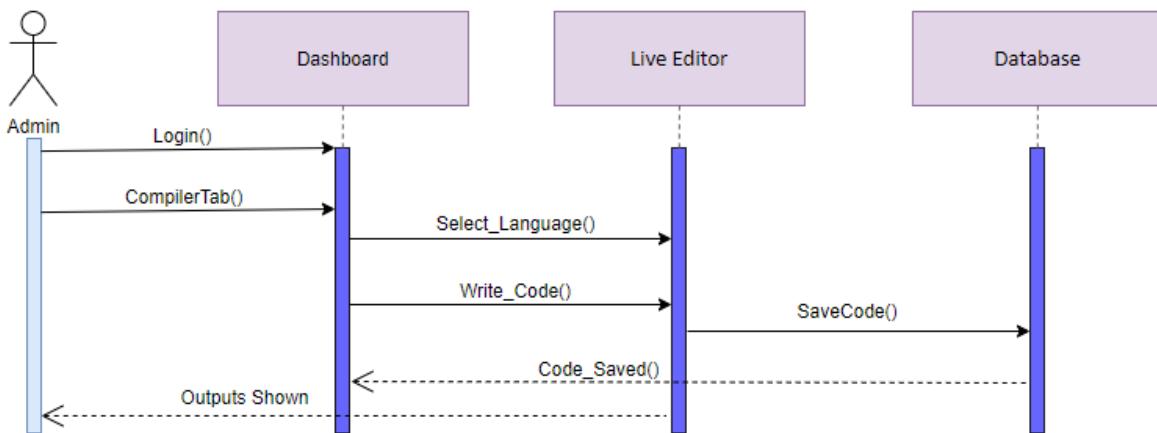


Figure 131 Sequence Diagram for Allow Registered user to Save Code in Repository

4.6.13 Edit User Code

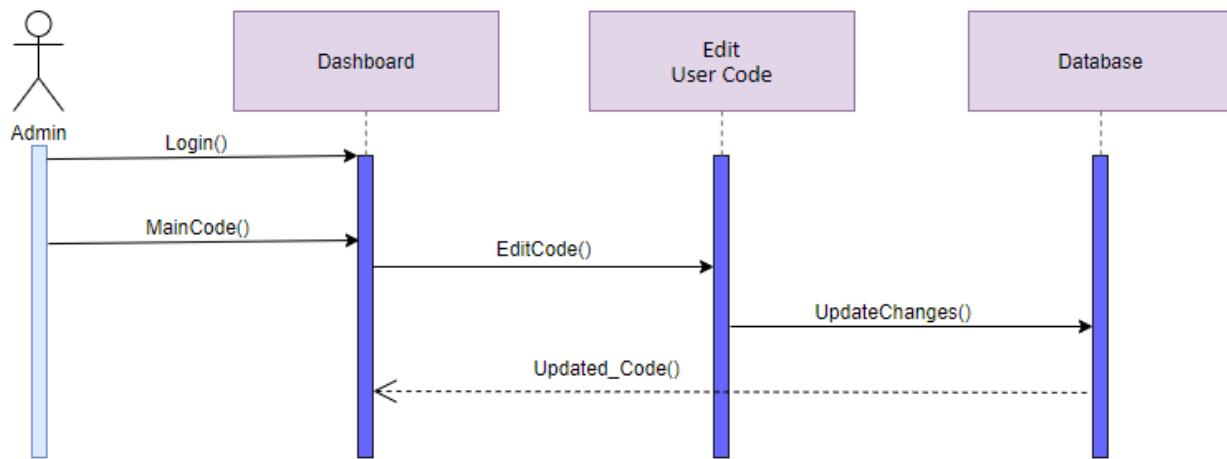


Figure 132 Sequence Diagram for Edit User Code

4.6.14 View User Code Details

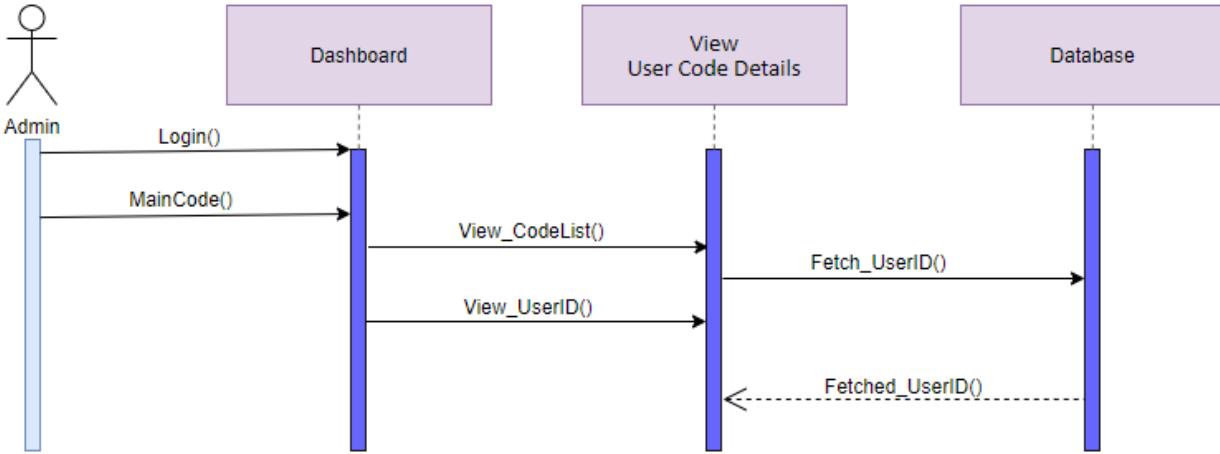


Figure 133 Sequence Diagram for View User Code Details

4.6.15 Add Code to Database

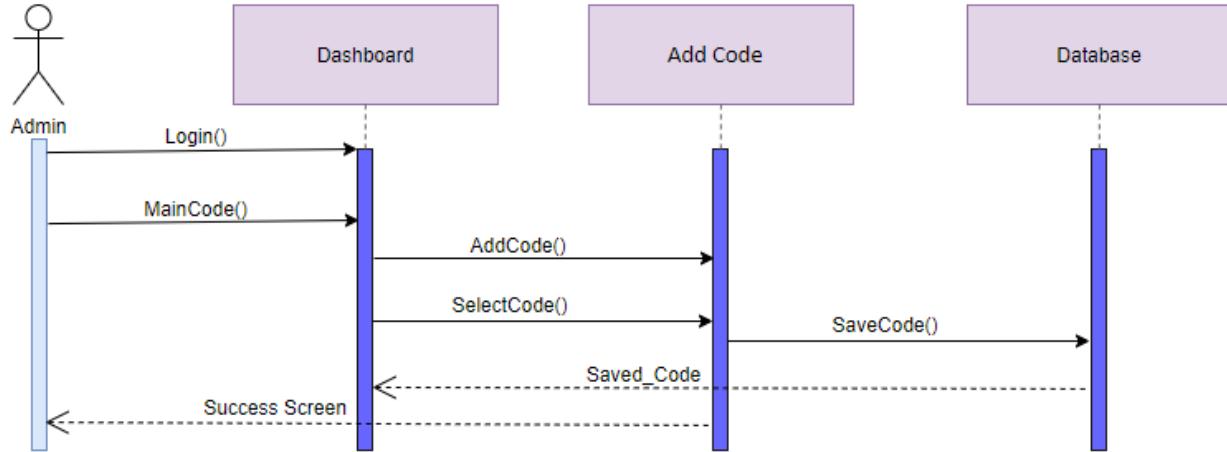


Figure 134 Sequence Diagram for Add Code to Database

4.6.16 View Code on Database

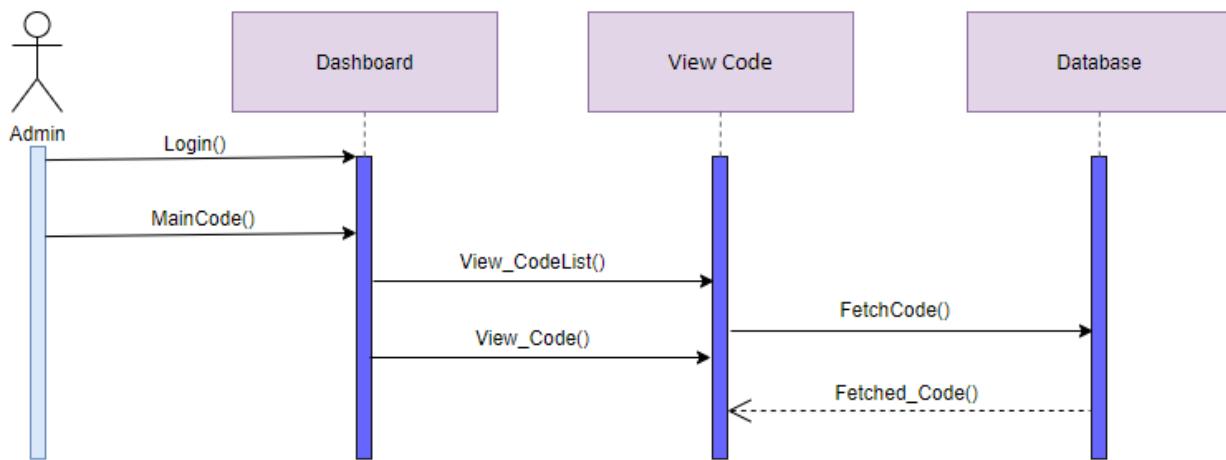


Figure 135 Sequence Diagram for View Code on Database

4.6.17 Delete Code

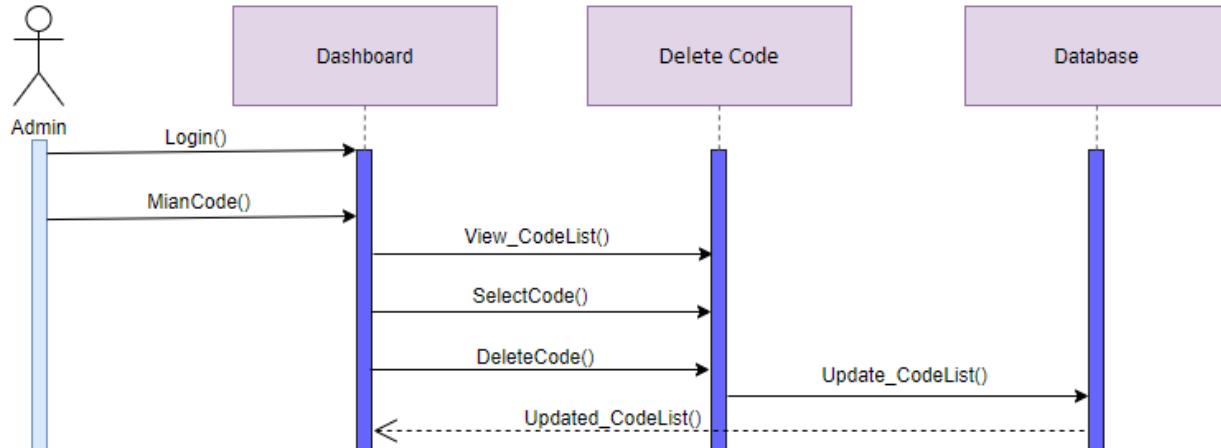


Figure 136 Sequence Diagram for Delete Code

4.6.18 Payment Gateway Integration

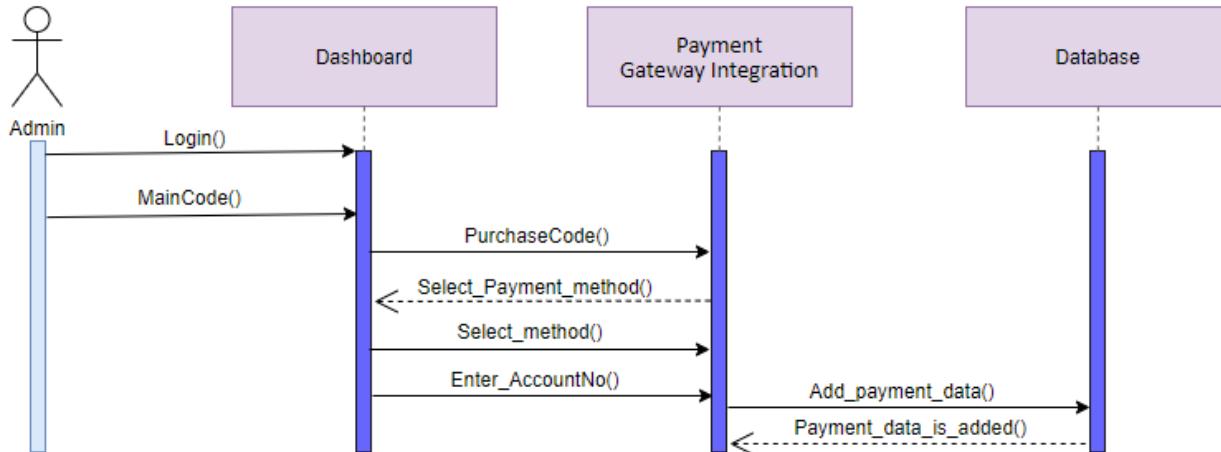


Figure 137 Sequence Diagram for Payment Gateway Integration

4.6.19 Sales and Purchase

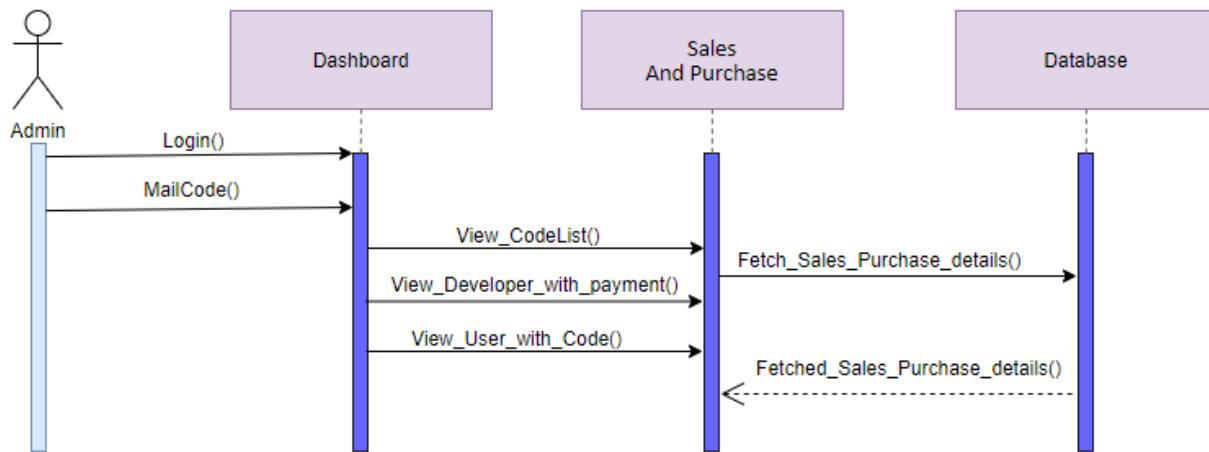


Figure 138 Sequence Diagram for Sales and Purchase

4.6.20 Multiple Currency

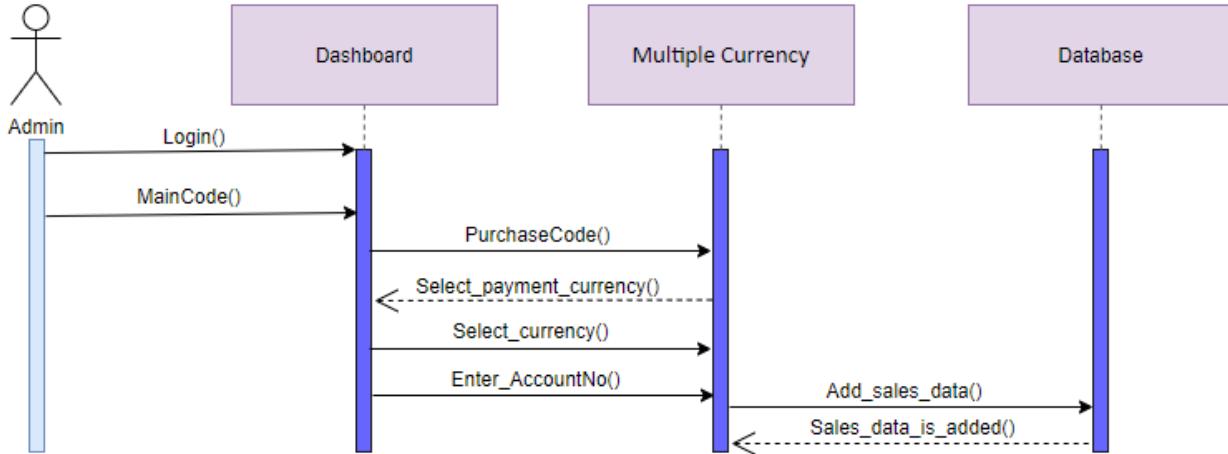


Figure 139 Sequence Diagram for Multiple Currency

4.6.21 Developer Sales Code

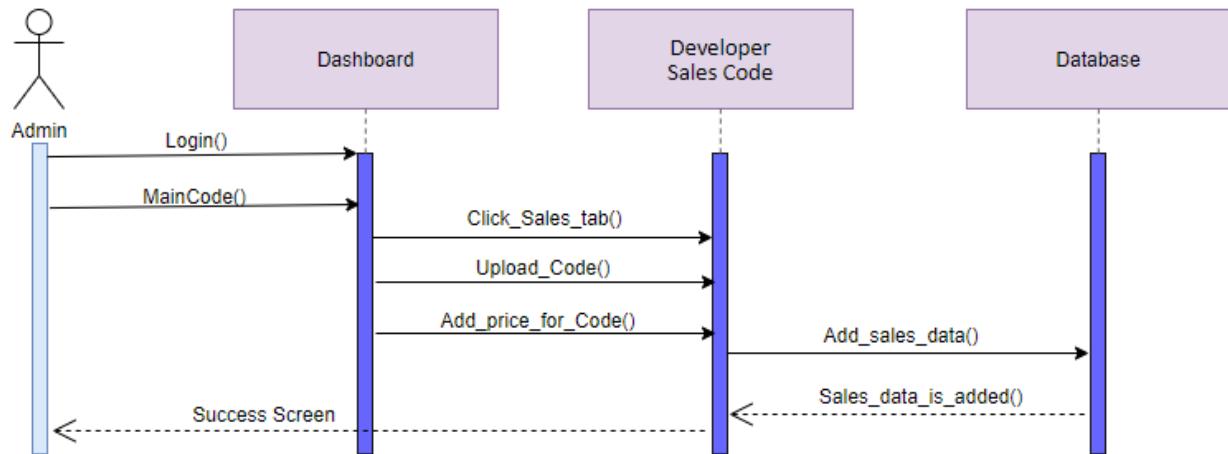


Figure 140 Sequence Diagram for Developer Sales Code

4.6.22 User Purchase Code

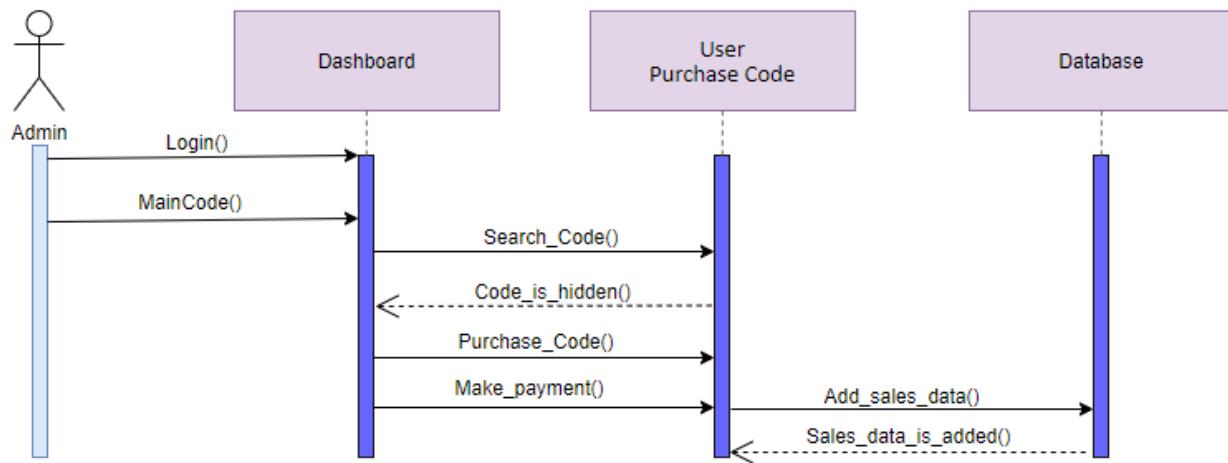


Figure 141 Sequence Diagram for User Purchase Code

4.6.23 Payment Report

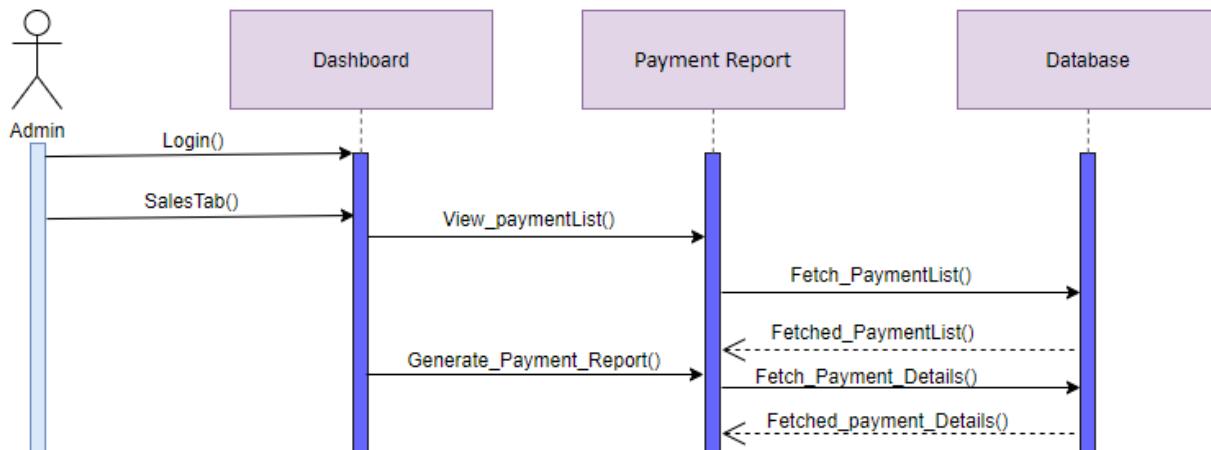


Figure 142 Sequence Diagram for Payment Report

4.6.24 Review on Codes

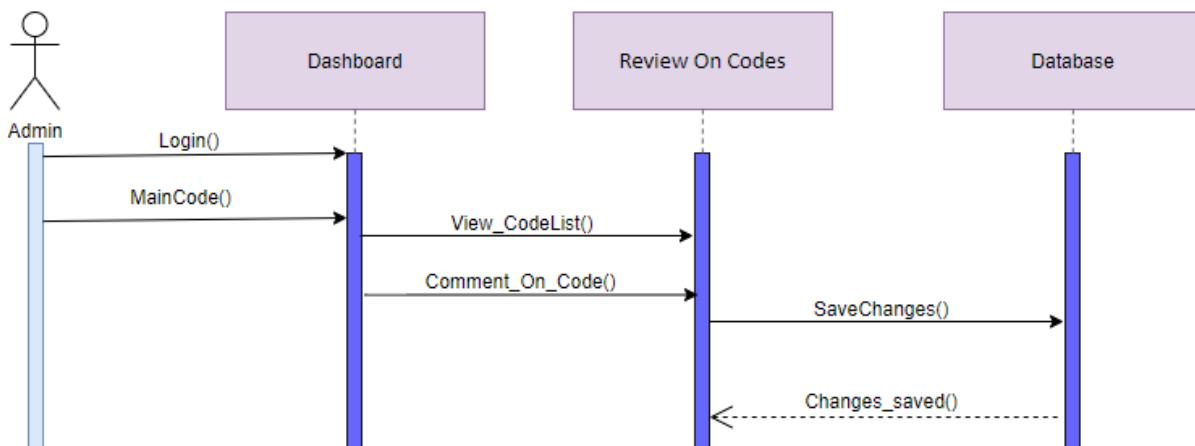


Figure 143 Sequence Diagram for Review on Codes

4.6.25 Ratings

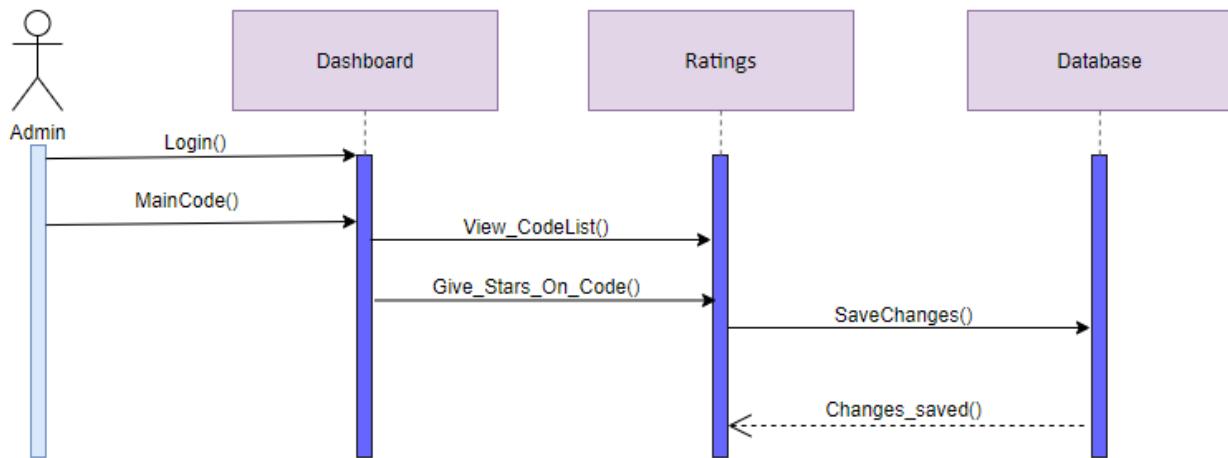


Figure 144 Sequence Diagram for Ratings

4.6.26 Ask Query

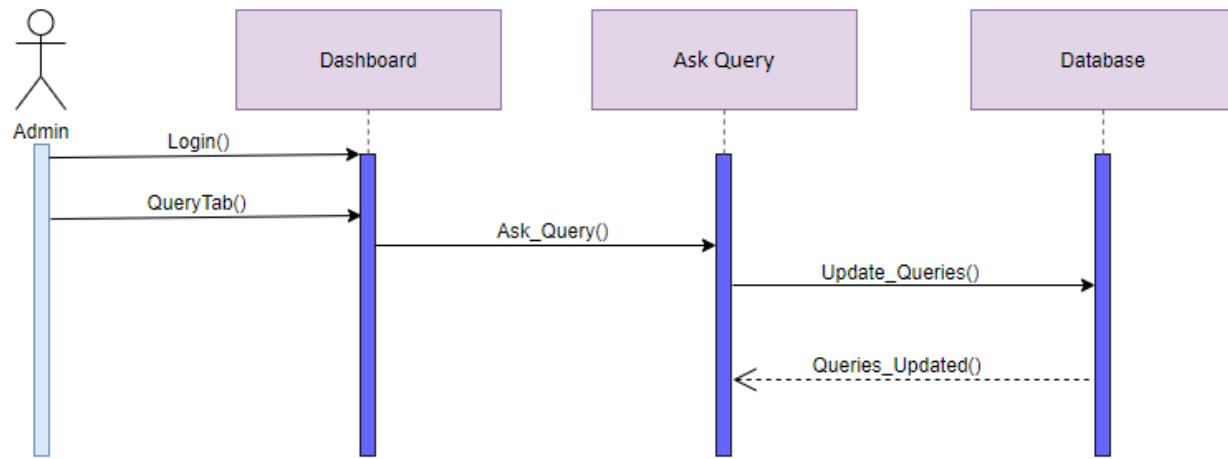


Figure 145 Sequence Diagram for Ask Query

4.6.27 Solution of Query

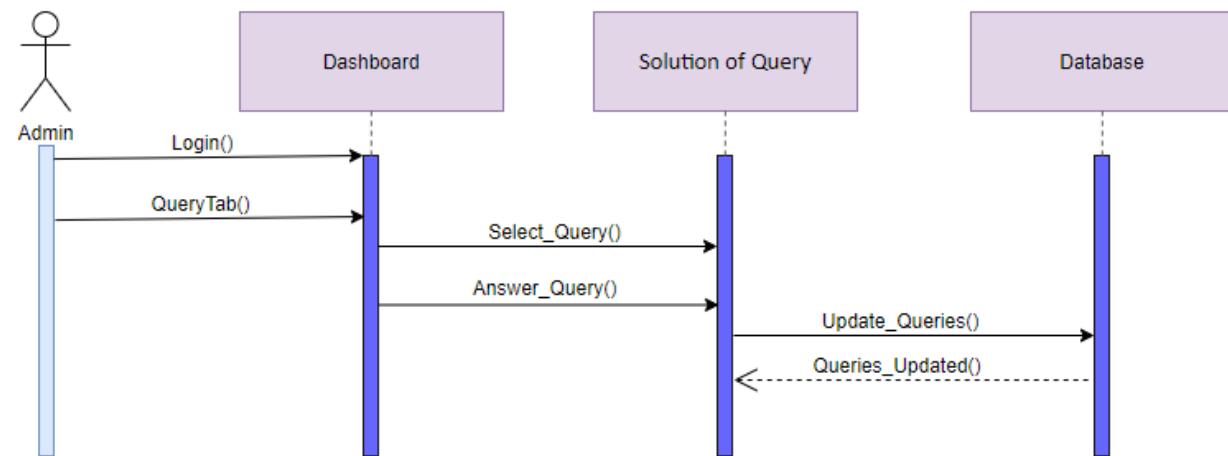


Figure 146 Sequence Diagram for Solution of Query

4.6.28 Admin Upload Codes

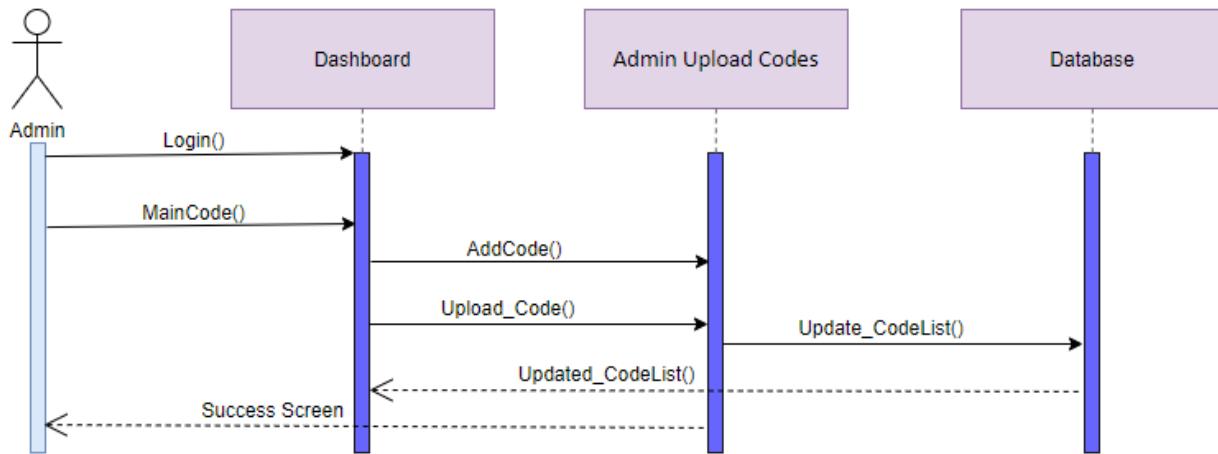


Figure 147 Sequence Diagram for Admin Upload Codes

4.6.29 User Upload Codes

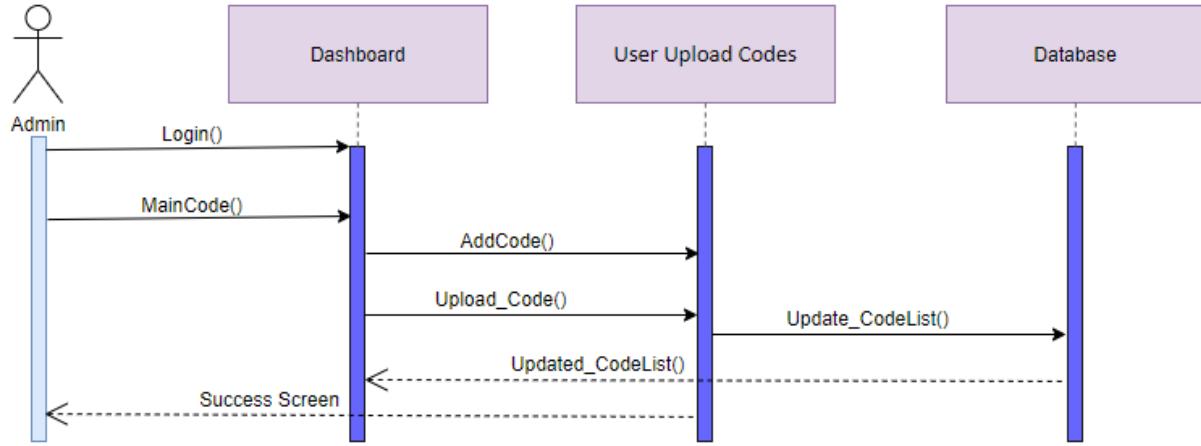


Figure 148 Sequence Diagram for User Upload Codes

4.6.30 User Rate Script

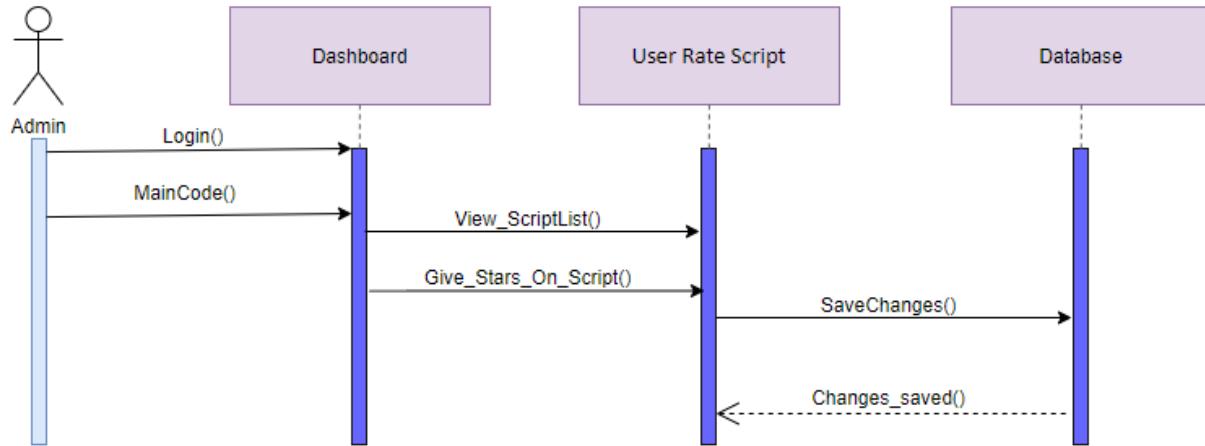


Figure 149 Sequence Diagram for User Rate Script

4.6.31 Allow User to Switch Role

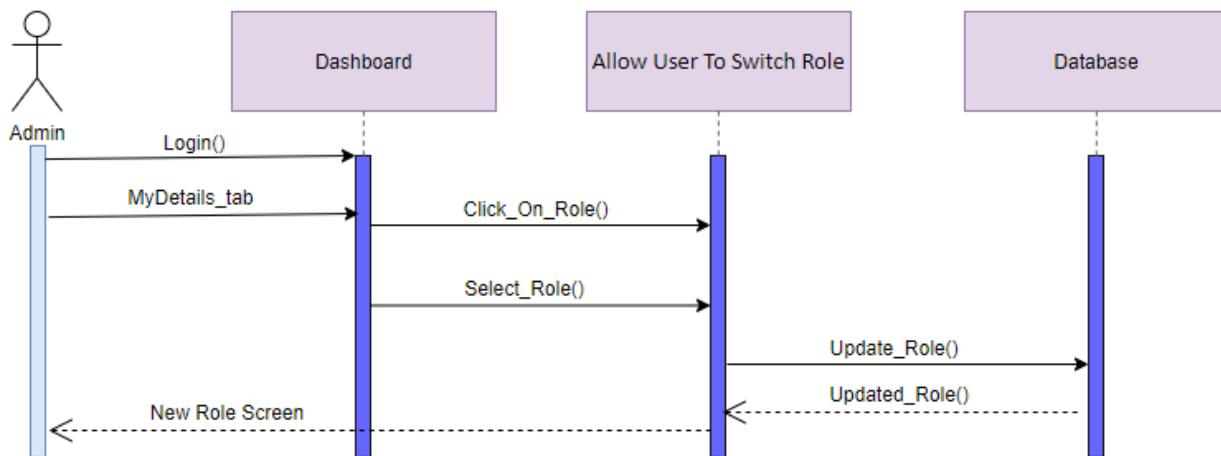


Figure 150 Sequence Diagram for Allow User to Switch Role

4.6.32 User Participate In Voting

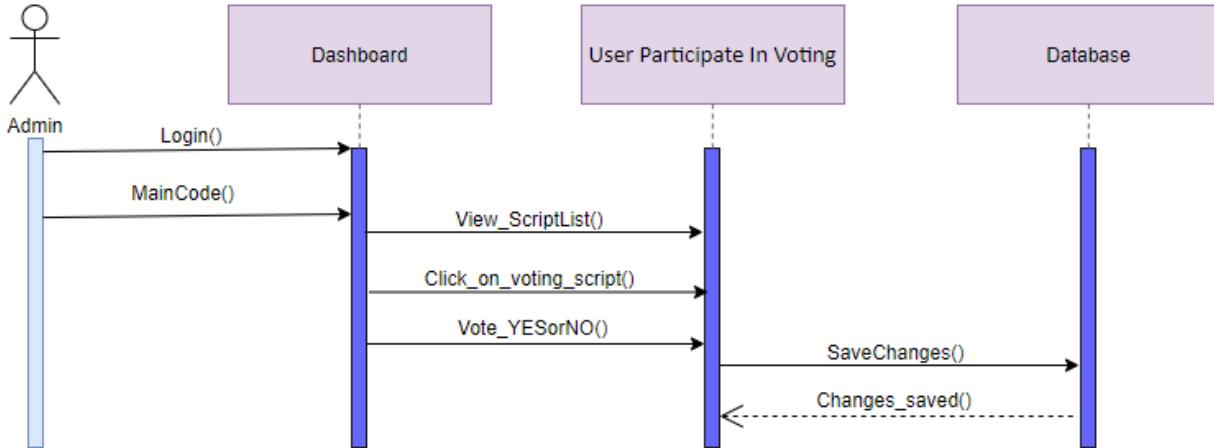


Figure 151 Sequence Diagram for User Participate In Voting

4.6.33 Upload Script

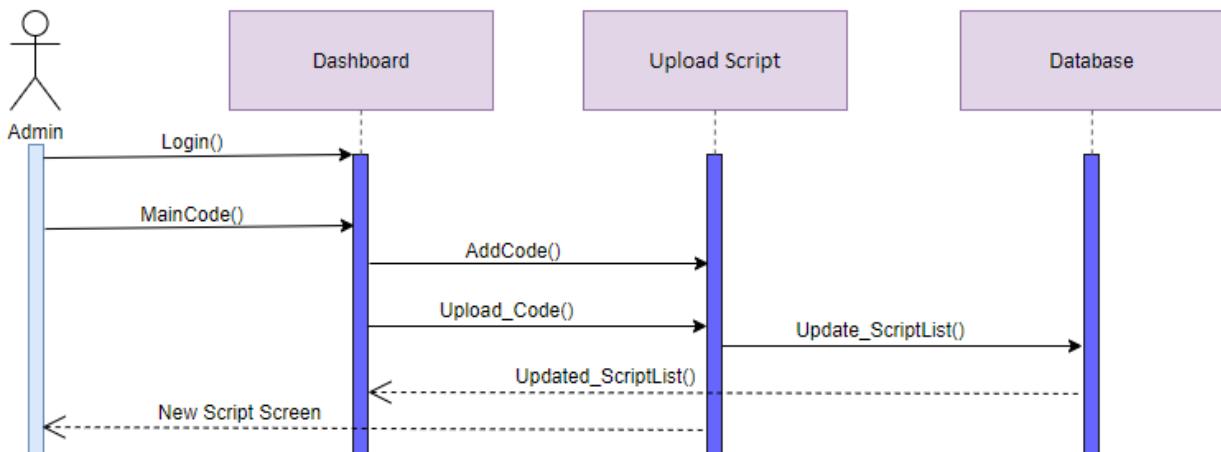


Figure 152 Sequence Diagram for Upload Script

4.6.34 Delete Script

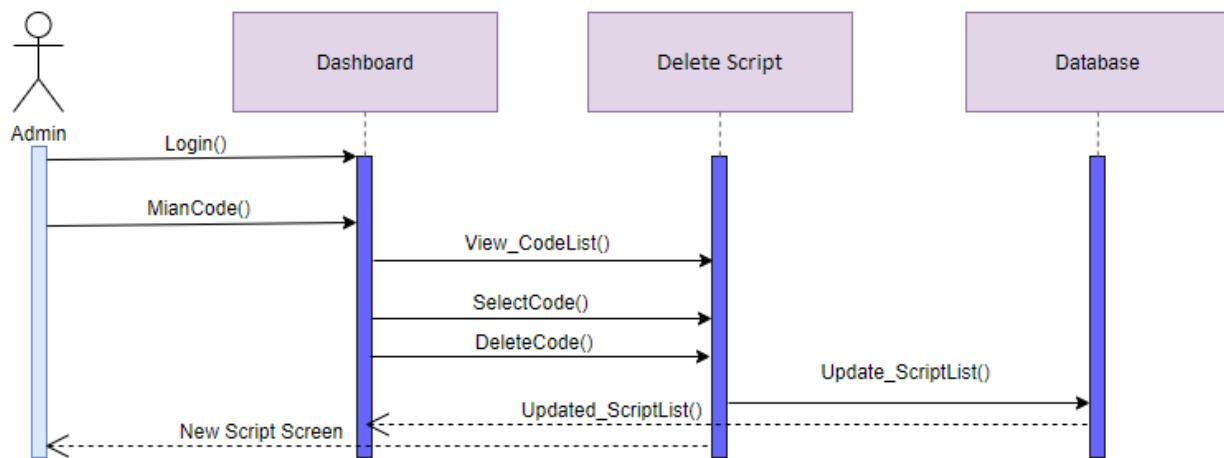


Figure 153 Sequence Diagram for Delete Script

4.6.35 Update Script

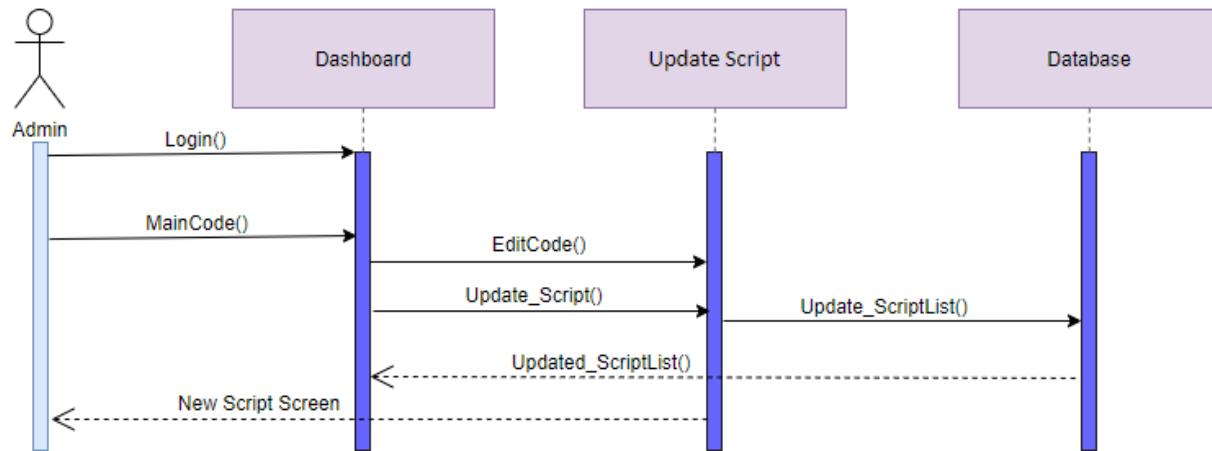


Figure 154 Sequence Diagram for Update Script

4.6.36 User Ask Query

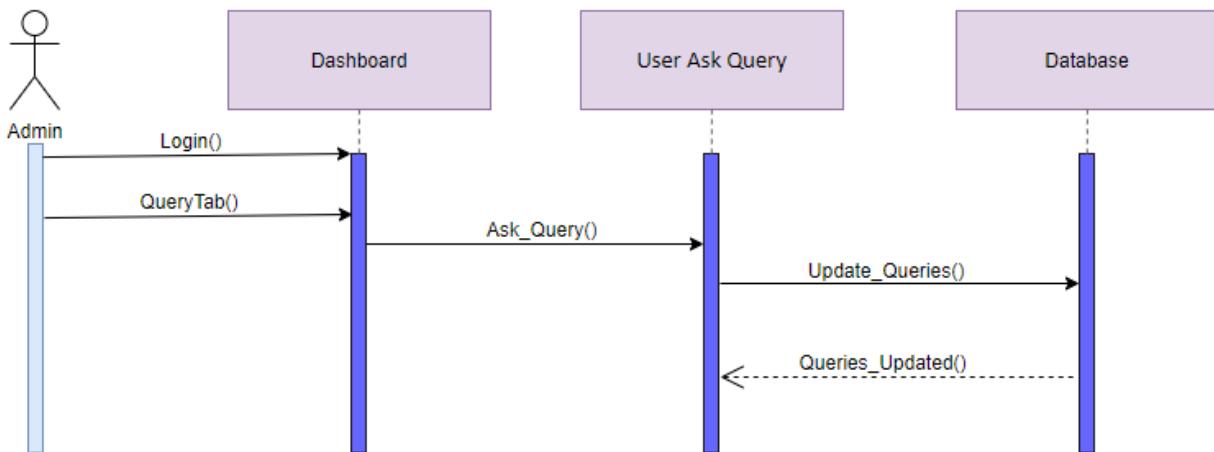


Figure 155 Sequence Diagram for User Ask Query

User Panel

4.6.37 Login

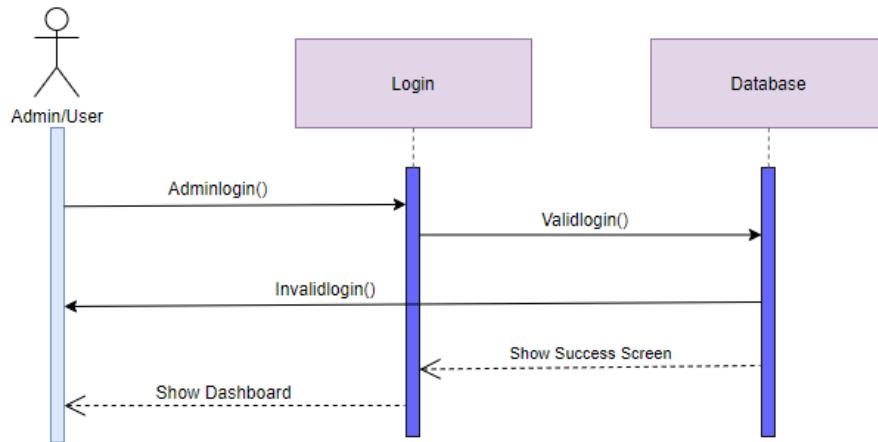


Figure 156 Sequence diagram for Login

4.6.38 Forget ID or Password

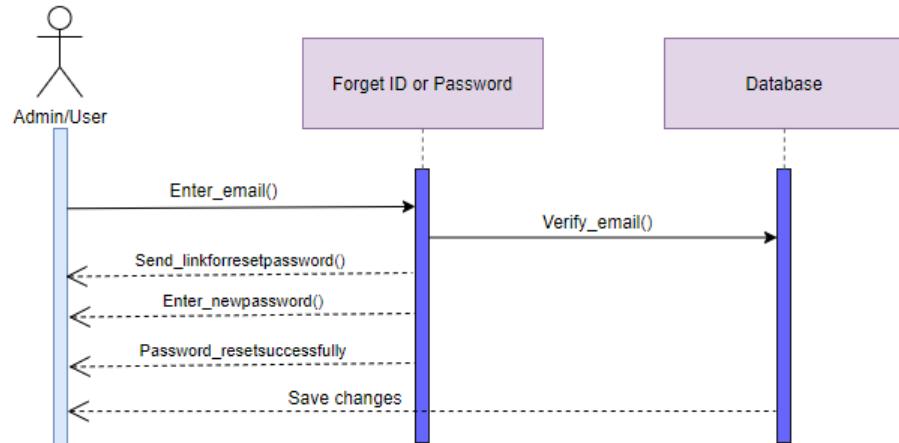


Figure 157 Sequence diagram for Forget ID or Password

4.6.39 Write Code on Live Editor

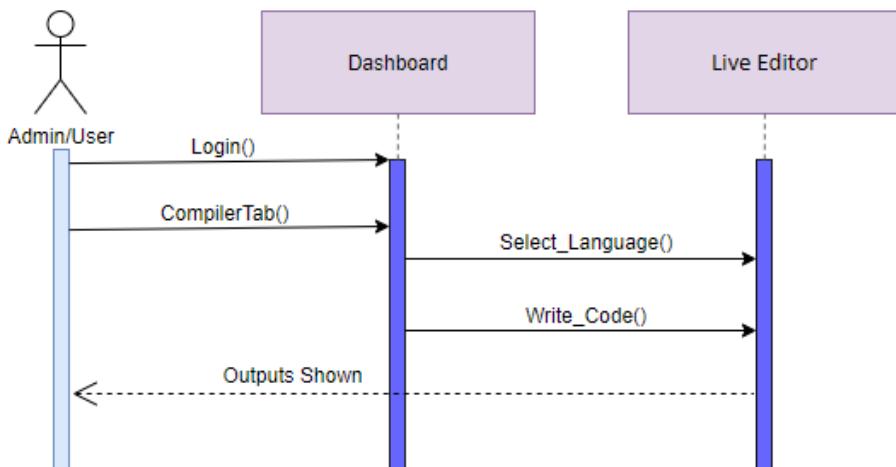


Figure 158 Sequence diagram for Write Code on Live Editor

4.6.40 save Code on Local Machine

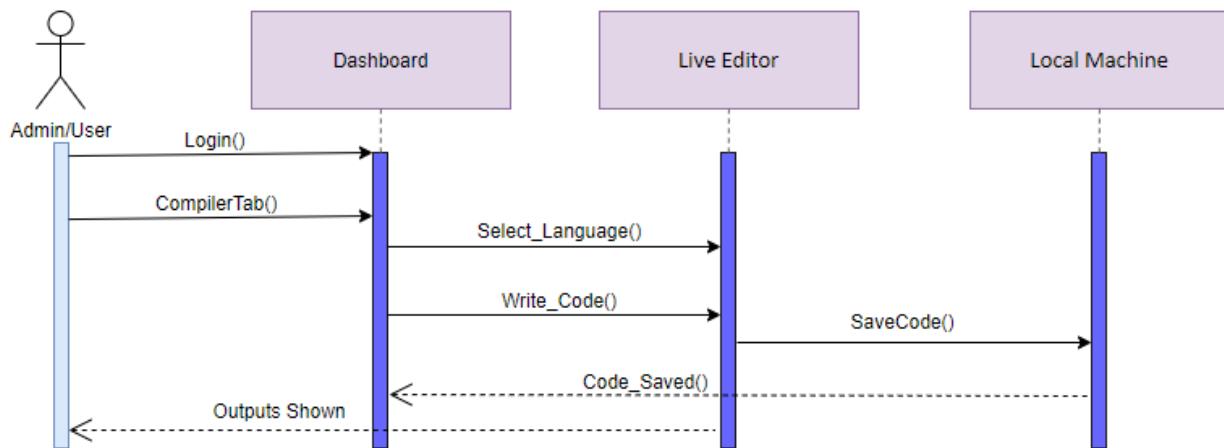


Figure 159 Sequence diagram for save Code on Local Machine

4.6.41 Allow Registered User to Save Code In Repository

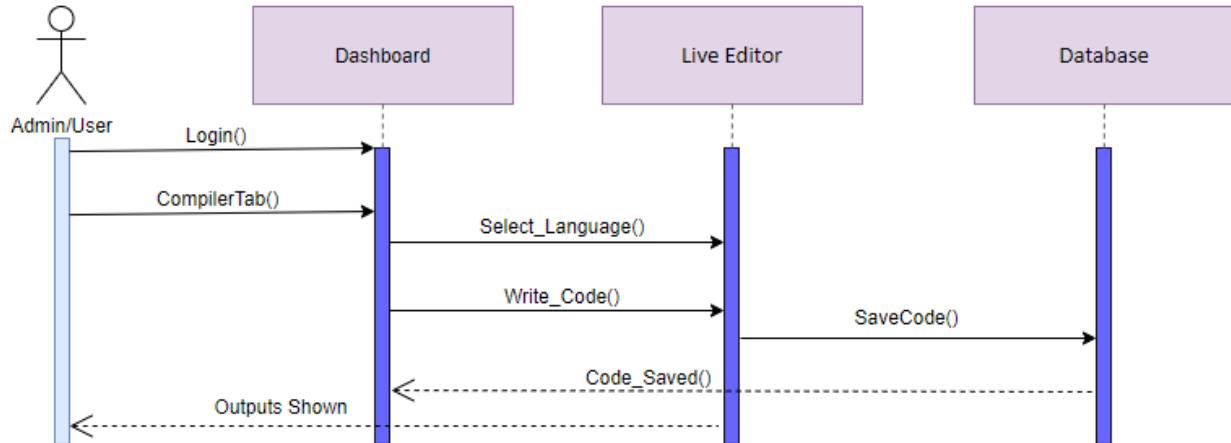


Figure 160 Sequence diagram for Allow Registered User to Save Code In Repository

4.6.42 Sales and Purchase

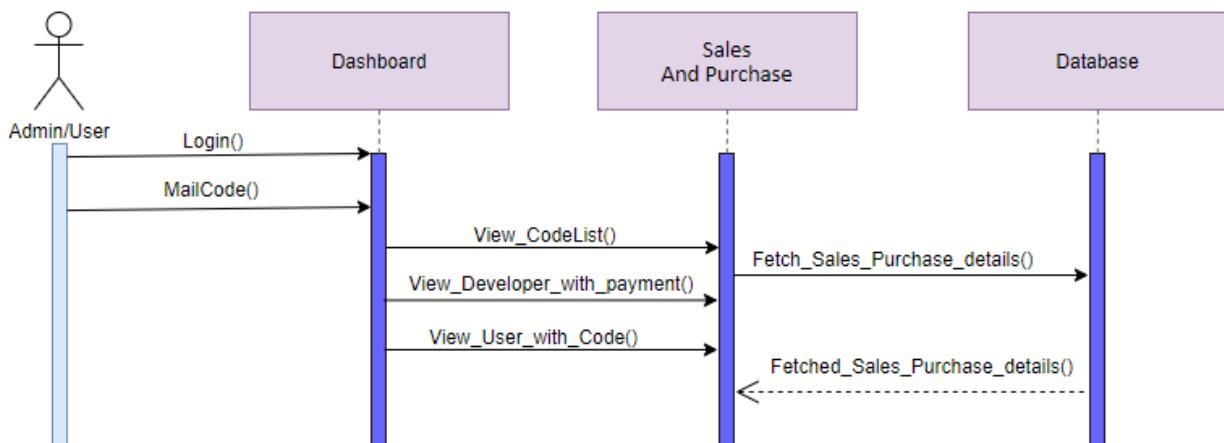


Figure 161 Sequence diagram for Sales and Purchase

4.6.43 Multiple Currency

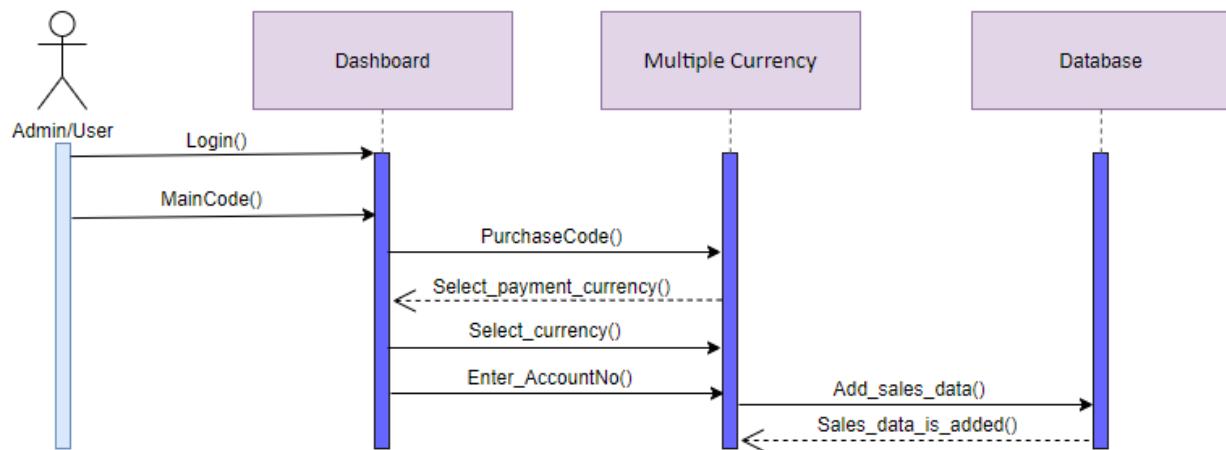


Figure 162 Sequence diagram for Multiple Currency

4.6.44 User Purchase Code

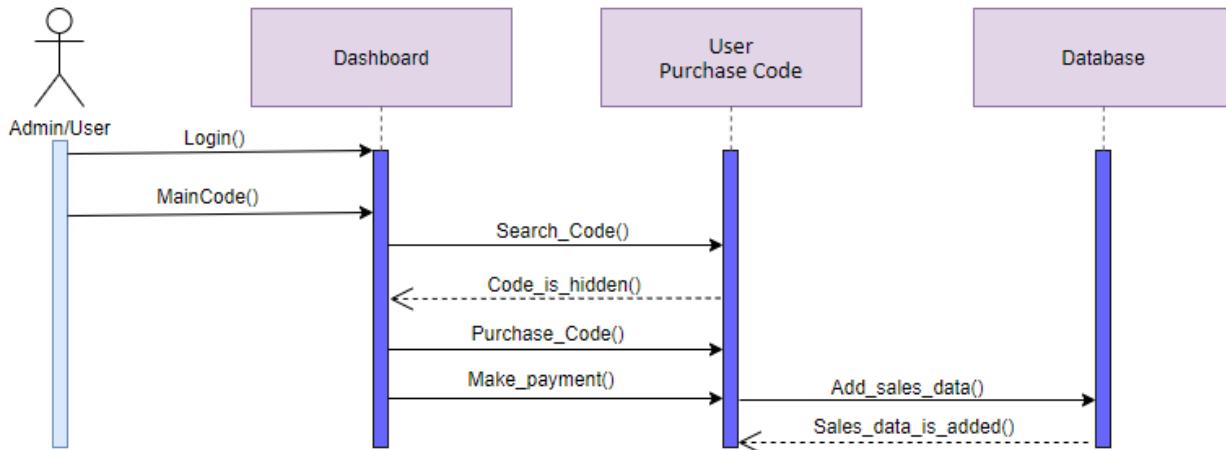


Figure 163 Sequence diagram for User Purchase Code

4.6.45 Payment Gateway Integration

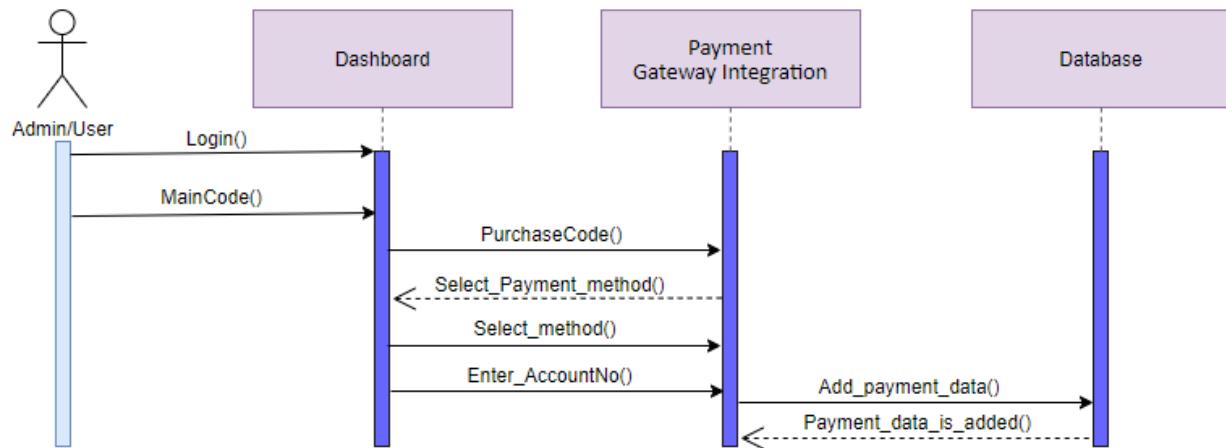


Figure 164 Sequence diagram for Payment Gateway Integration

4.6.46 Review on Codes

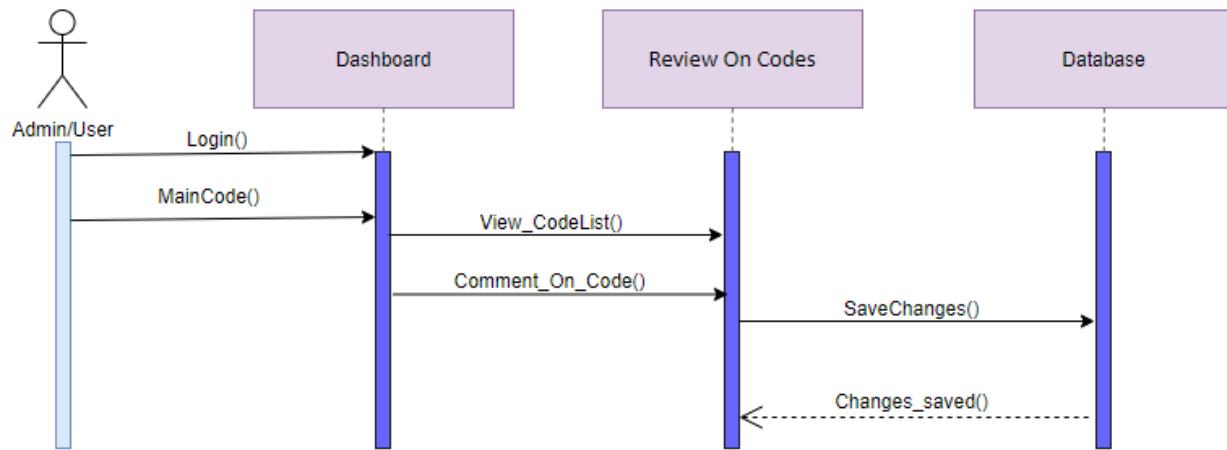


Figure 165 Sequence diagram for Review on Codes

4.6.47 Ratings

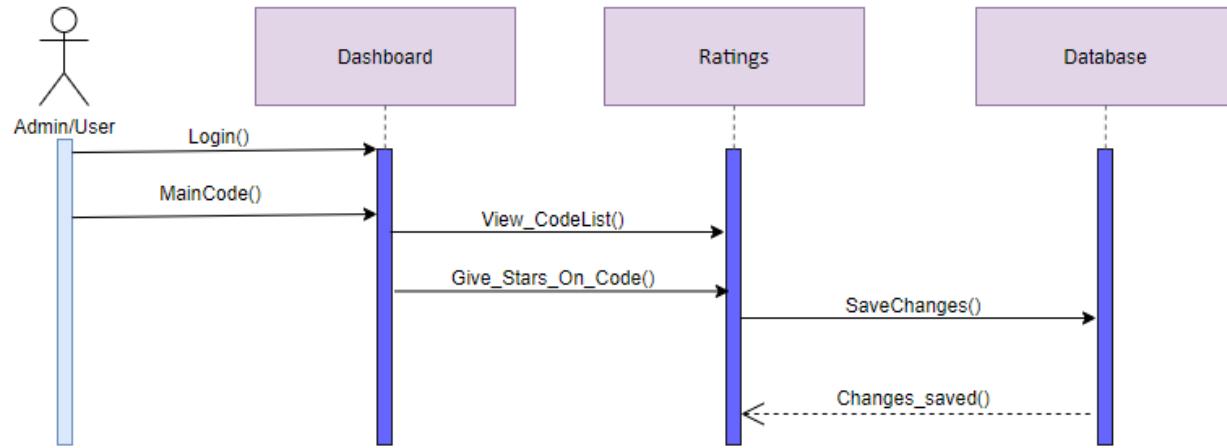


Figure 166 Sequence diagram for Ratings

4.6.48 Ask Query

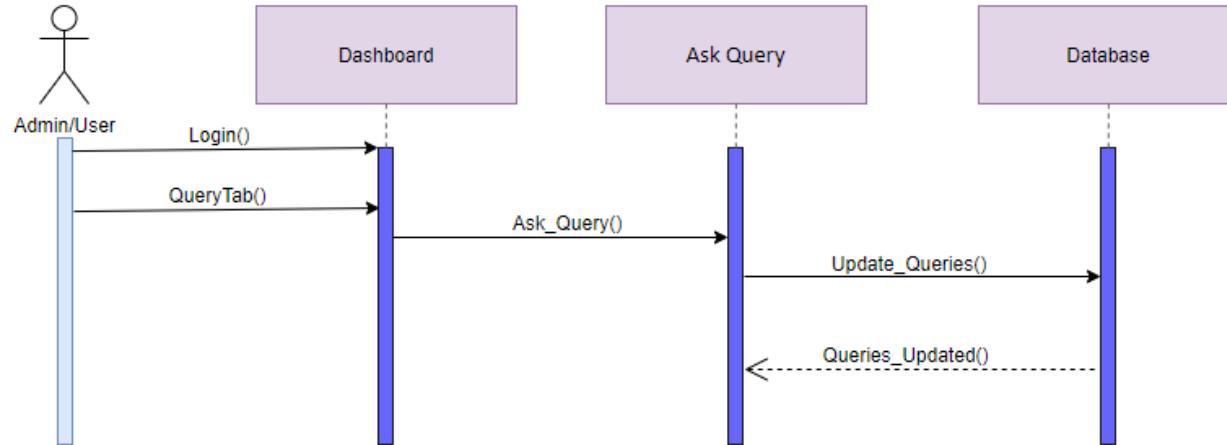


Figure 167 Sequence diagram for Ask Query

4.6.49 Solution of Query

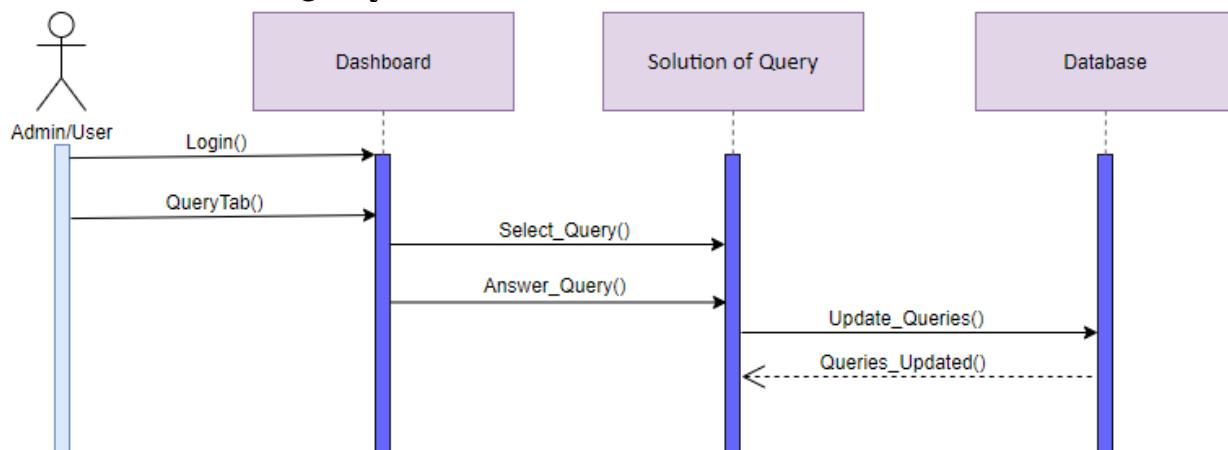


Figure 168 Sequence diagram for Solution of Query

4.6.50 User Upload Codes

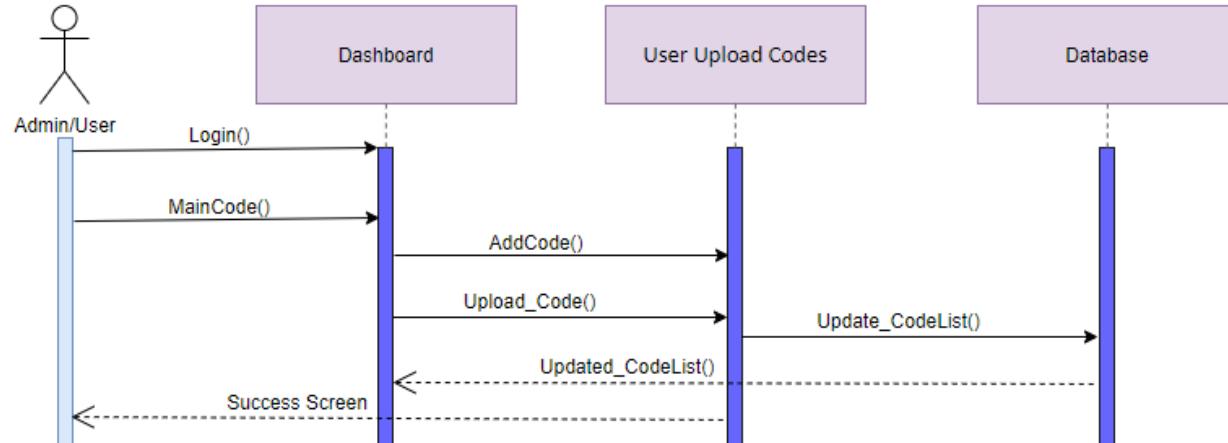


Figure 169 Sequence diagram for User Upload Codes

4.6.51 User Rate Script

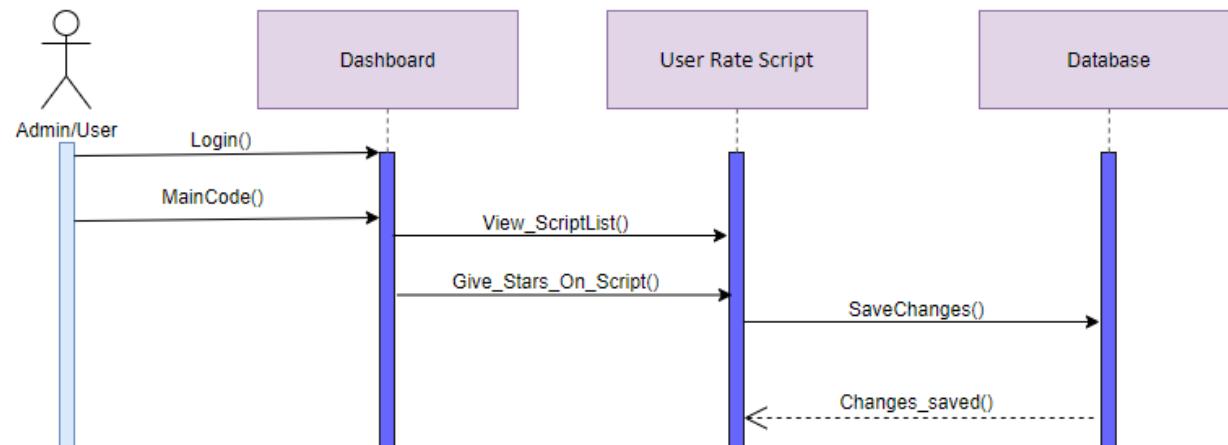


Figure 170 Sequence diagram for User Rate Script

4.6.52 Allow User to Switch Role

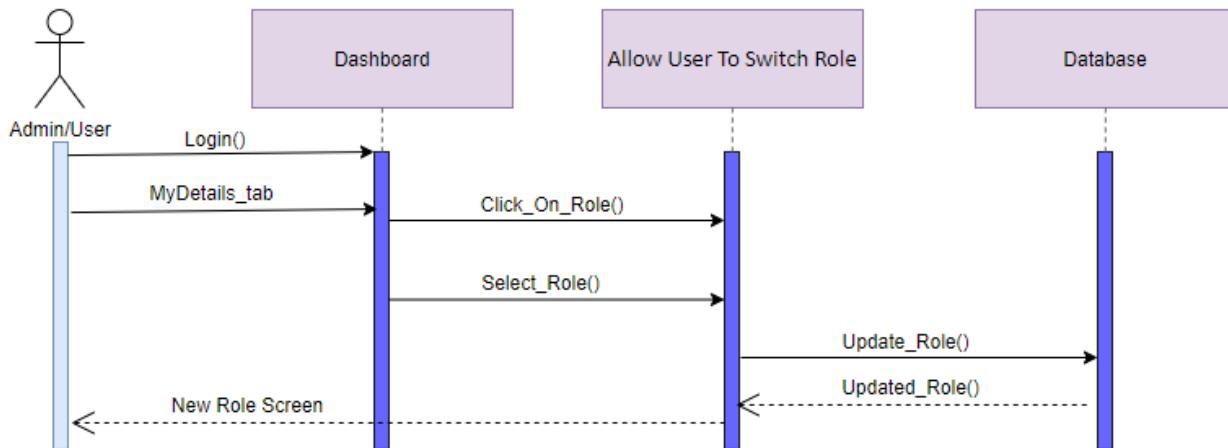


Figure 171 Sequence diagram for Allow User to Switch Role

4.6.53 User Participate In Voting

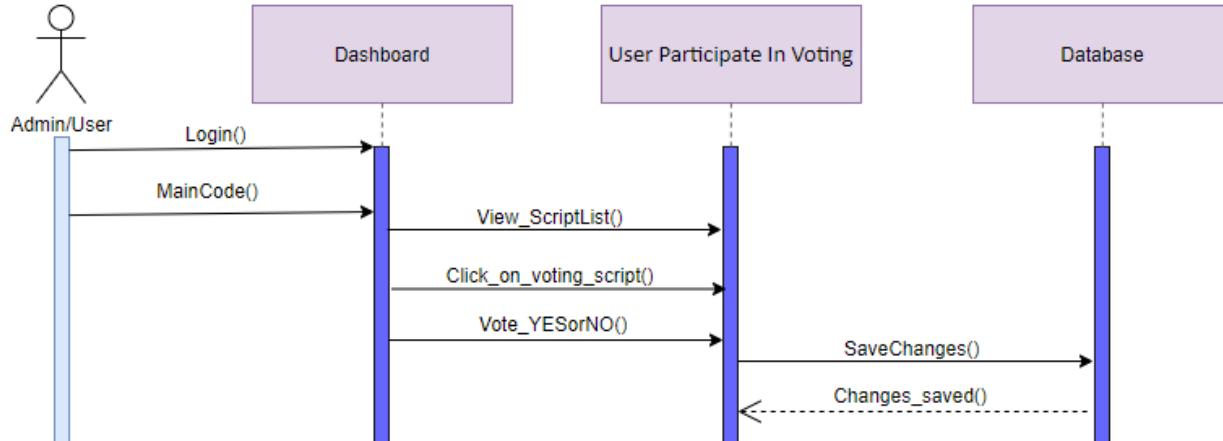


Figure 172 Sequence diagram for User Participate In Voting

4.6.54 Delete Script

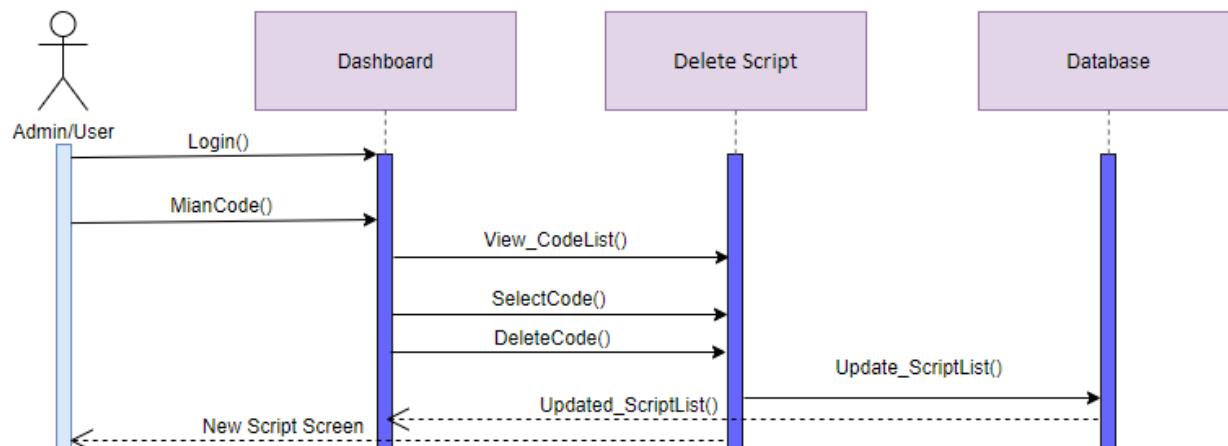


Figure 173 Sequence diagram for Delete Script

4.6.55 Update Script

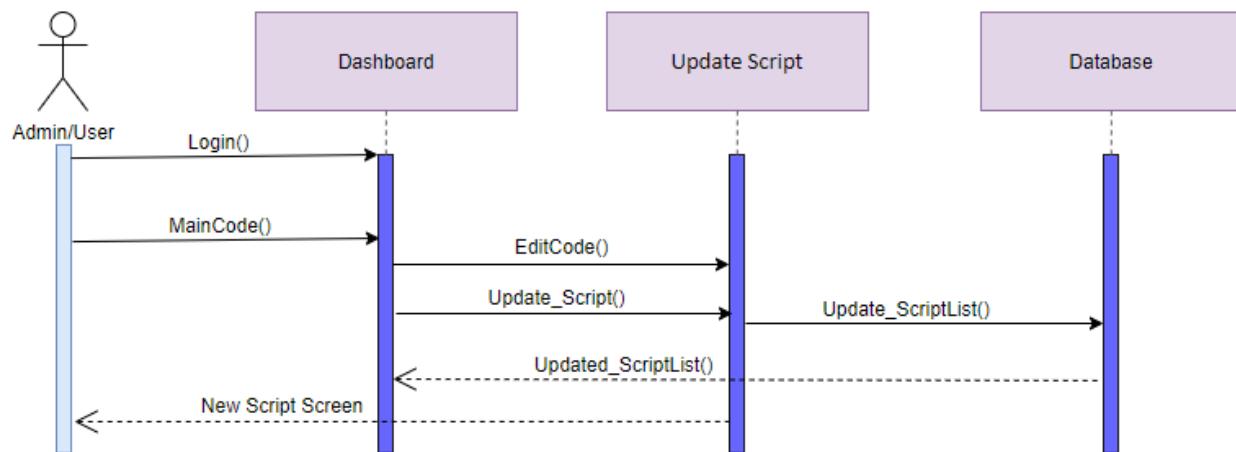


Figure 174 Sequence diagram for Update Script

4.7 Collaboration Diagram

4.7.1 Login For Admin

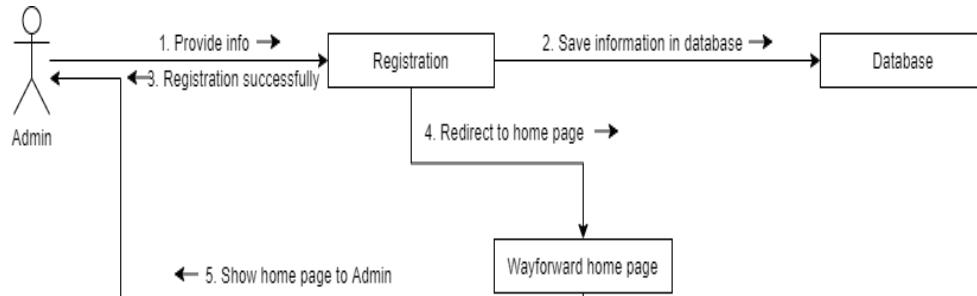


Figure 175 Collaboration Diagram for Admin Login

For User

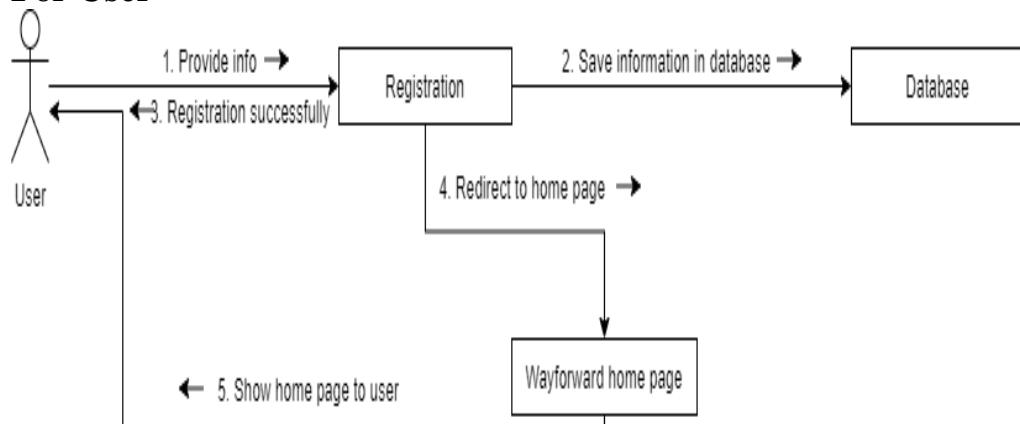


Figure 176 Collaboration Diagram for User Login

4.7.2 Forget ID & Password

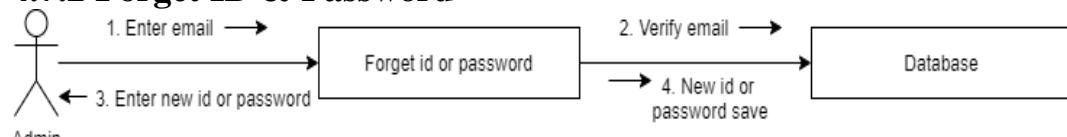


Figure 177 Collaboration Diagram for Forget ID & Password

4.7.3 Register New User

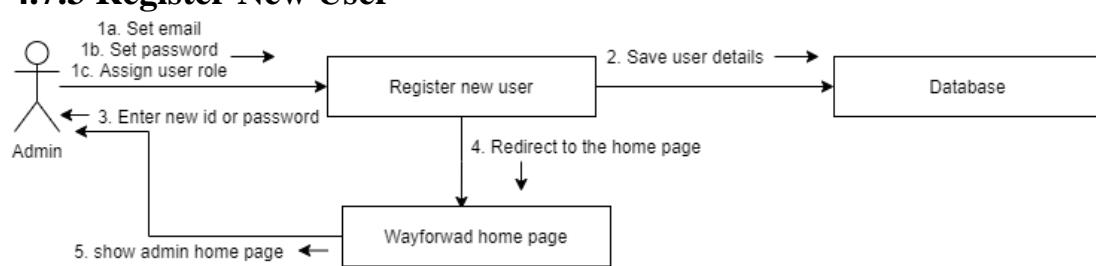


Figure 178 Collaboration Diagram for Register New User

4.7.4 Approve User Registration

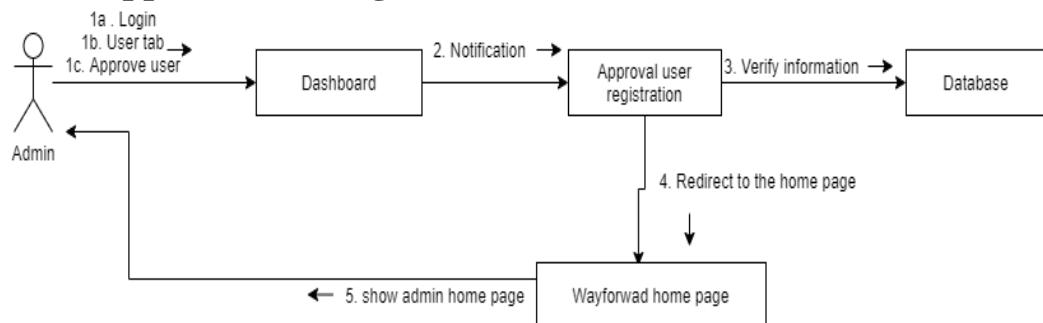


Figure 179 Collaboration Diagram for Approve User Registration

4.7.5 Block User

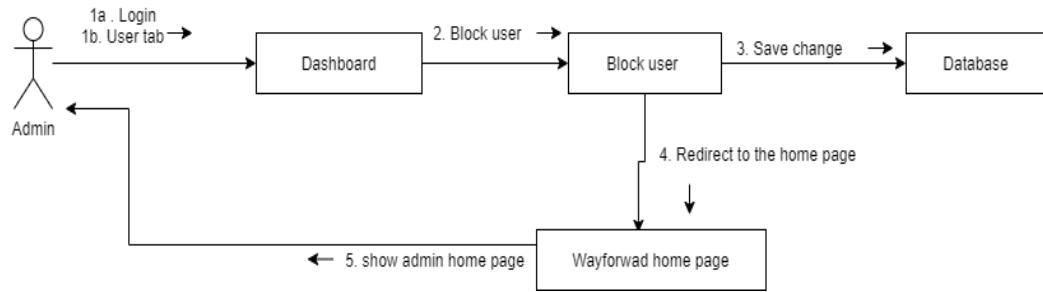


Figure 180 Collaboration Diagram for Block User

4.7.6 Assign Specific Role

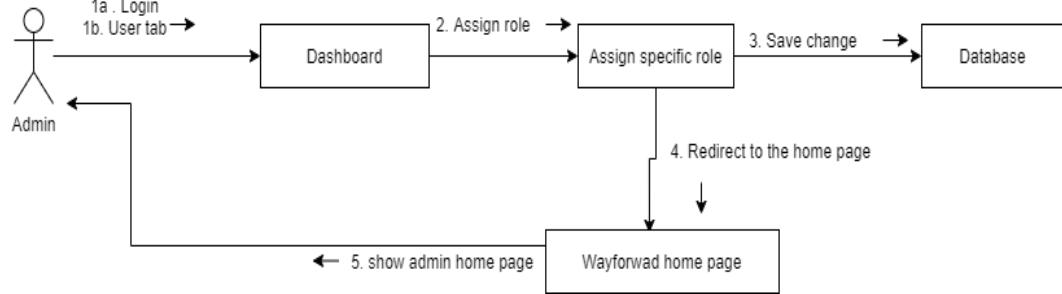


Figure 181 Collaboration Diagram for Assign Specific Role

4.7.7 Approve User's Queries

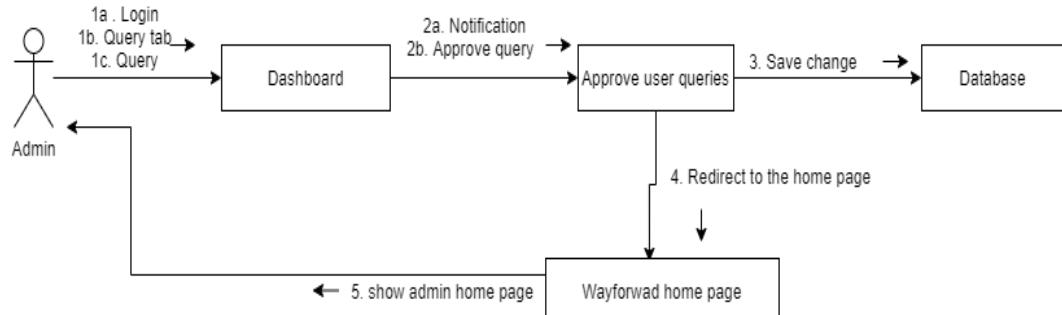


Figure 182 Collaboration Diagram for Approve User's Queries

4.7.8 Remove User's Queries

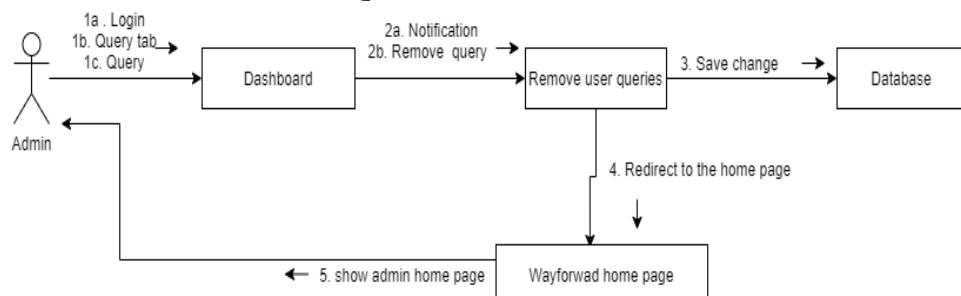


Figure 183 Collaboration Diagram for Remove User's Queries

4.7.9 Approve Developer' Script

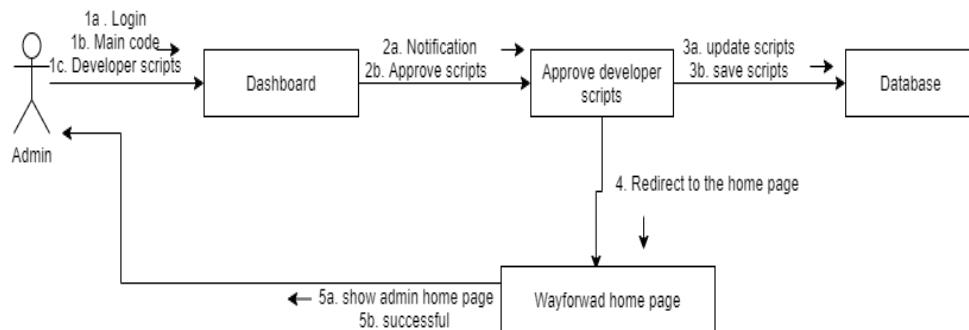


Figure 184 Collaboration Diagram for Approve Developer' Script

4.7.10 Write Code on Live Editor

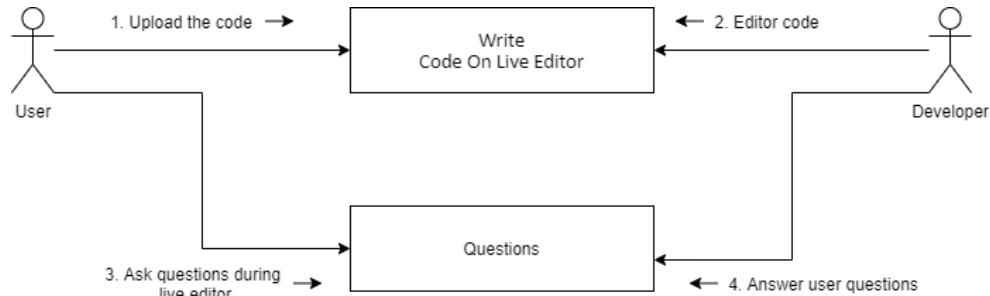


Figure 185 Collaboration Diagram for Write Code on Live Editor

4.7.11 save Code on Local Machine

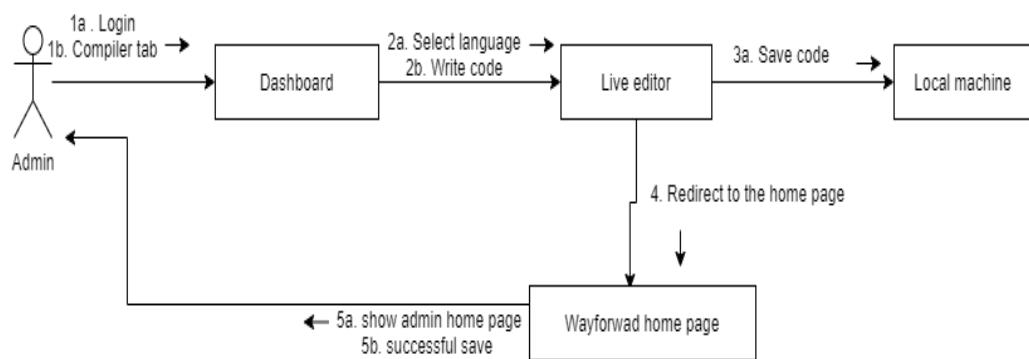


Figure 186 Collaboration Diagram for save Code on Local Machine

4.7.12 Registered User Save Code on Repository

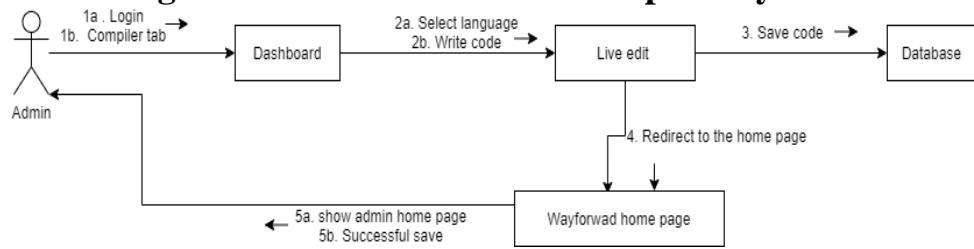


Figure 187 Collaboration Diagram for Registered User save Code on Repository

4.7.13 Edit User Code

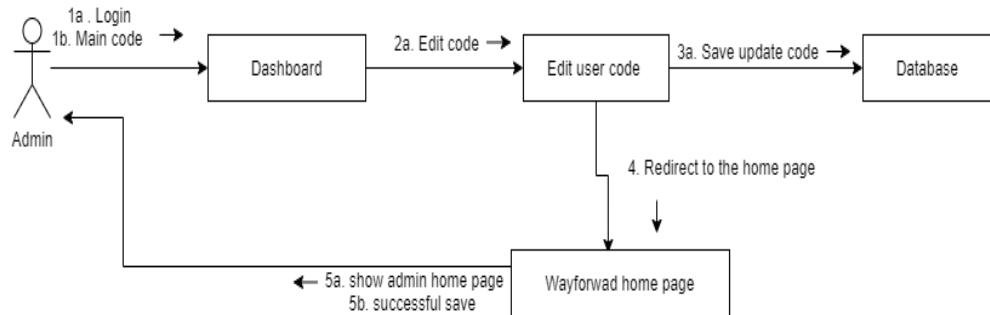


Figure 188 Collaboration Diagram for Edit User Code

4.7.14 View User Code Details

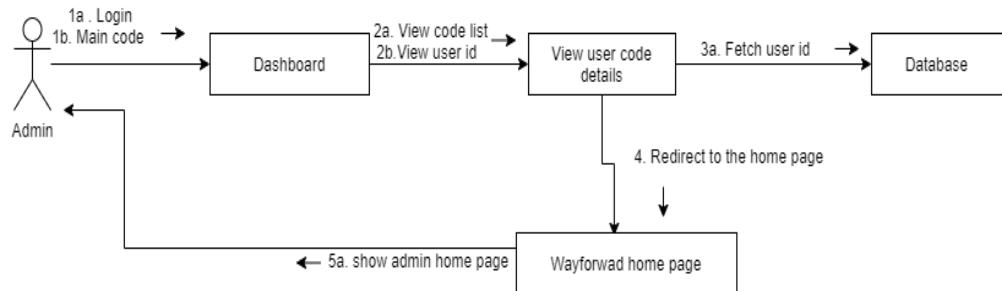


Figure 189 Collaboration Diagram for View User Code Details

4.7.15 Add Code to Database

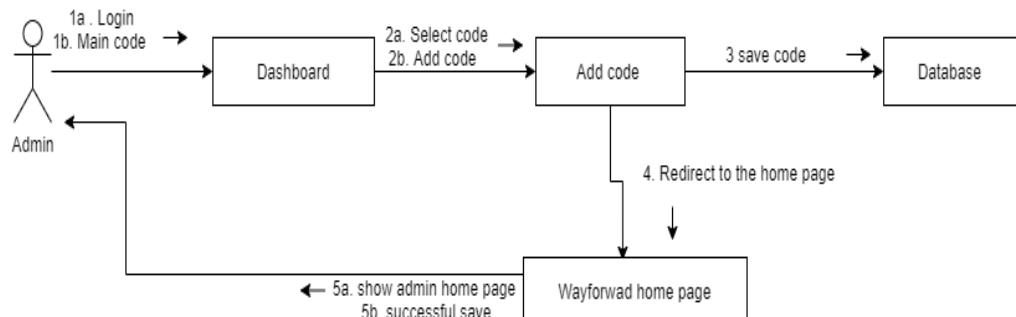


Figure 190 Collaboration Diagram for Add Code to Database

4.7.16 View Code on Database

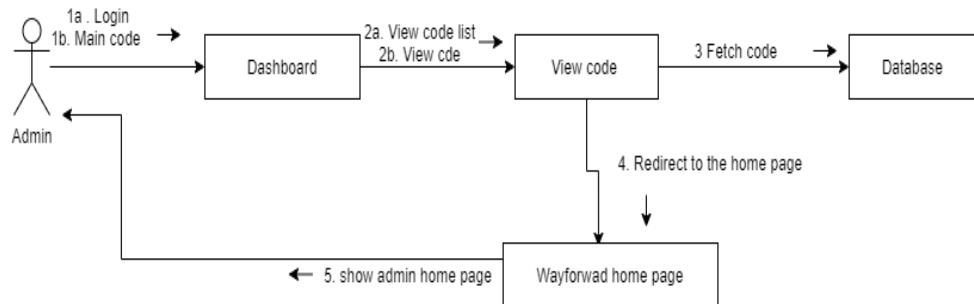


Figure 191 Collaboration Diagram for View Code on Database

4.7.17 Delete Code

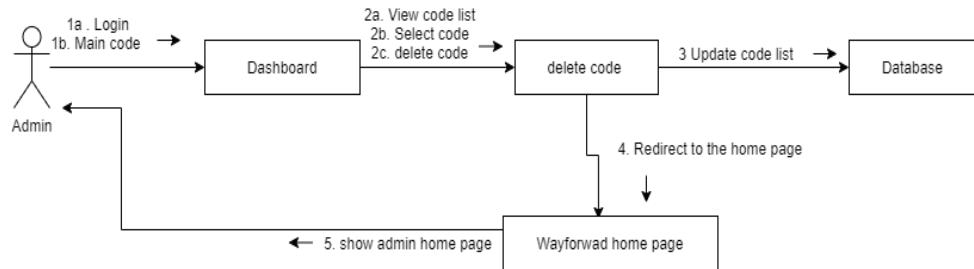


Figure 192 Collaboration Diagram for Delete Code

4.7.18 Payment Gateway Integration

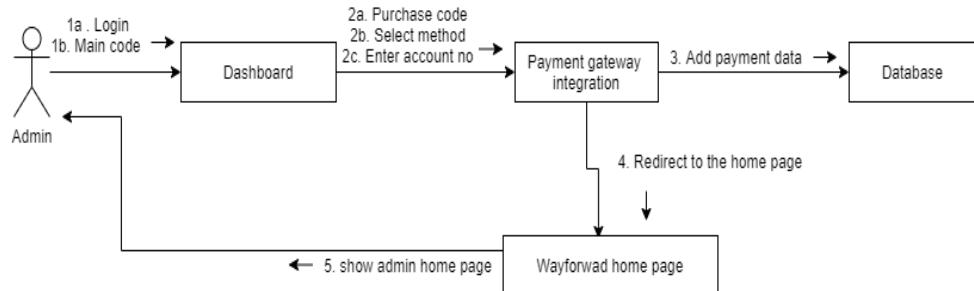


Figure 193 Collaboration Diagram for Payment Gateway Integration

4.7.19 Sales and Purchase

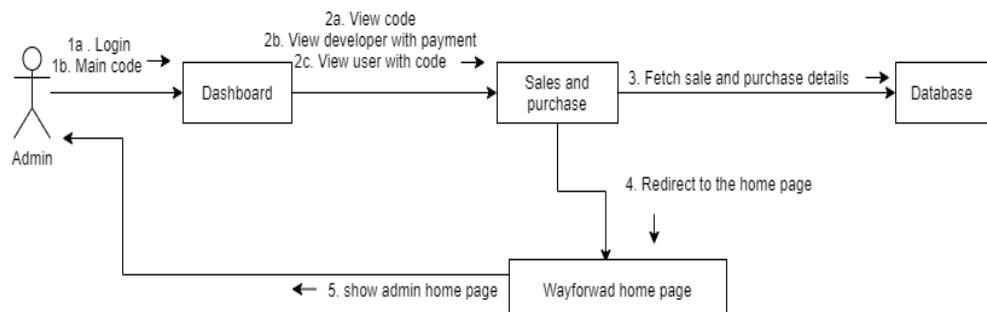


Figure 194 Collaboration Diagram for Sales and Purchase

4.7.20 Multiple Currency

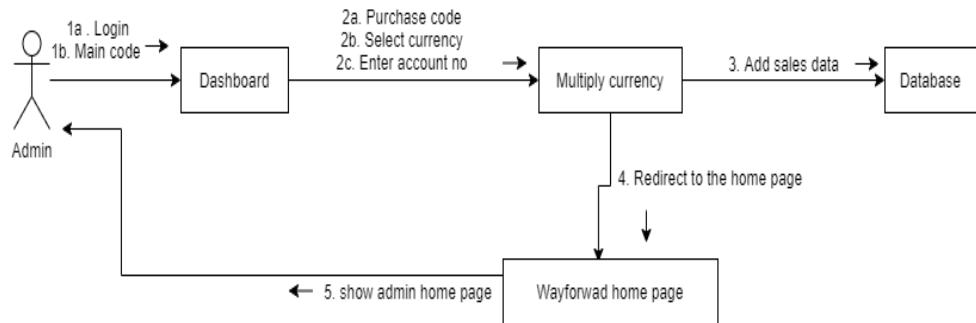


Figure 195 Collaboration Diagram for Multiple Currency

4.7.21 Developer Sales Code

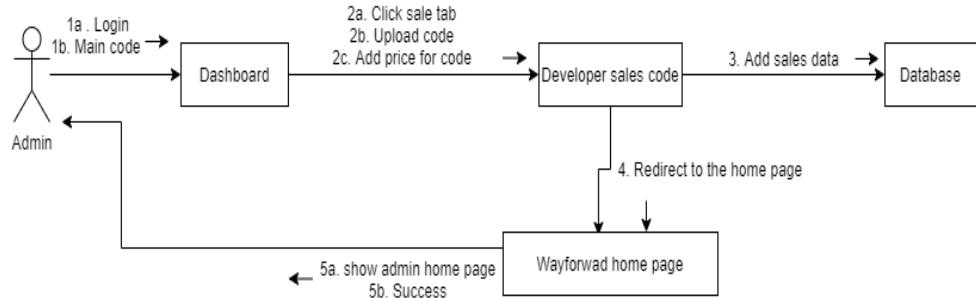


Figure 196 Collaboration Diagram for Developer Sales Code

4.7.22 User Purchase Code

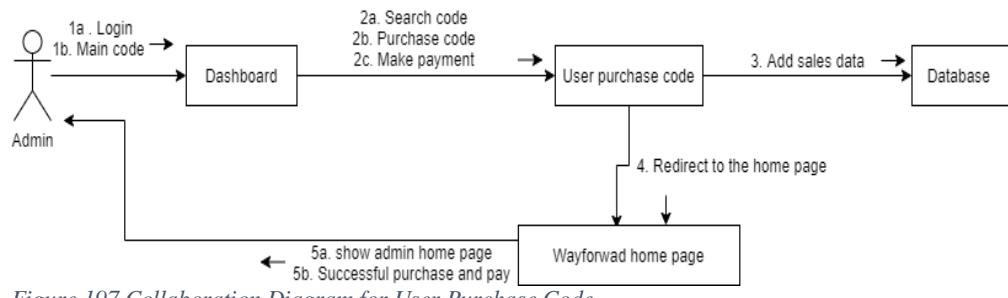


Figure 197 Collaboration Diagram for User Purchase Code

4.7.23 Payment Report

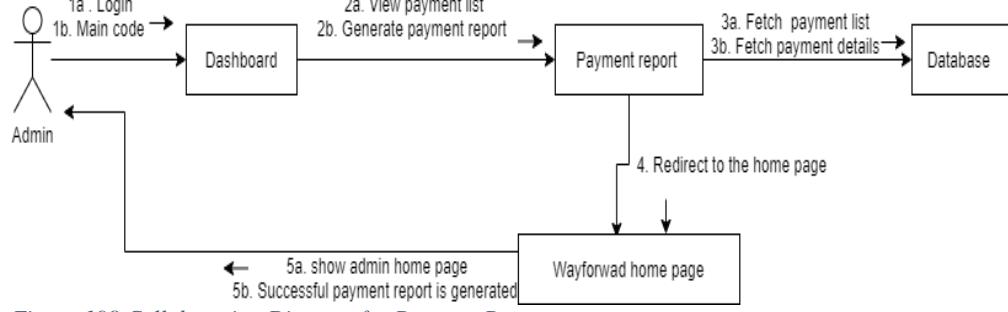


Figure 198 Collaboration Diagram for Payment Report

4.7.24 Review on Code

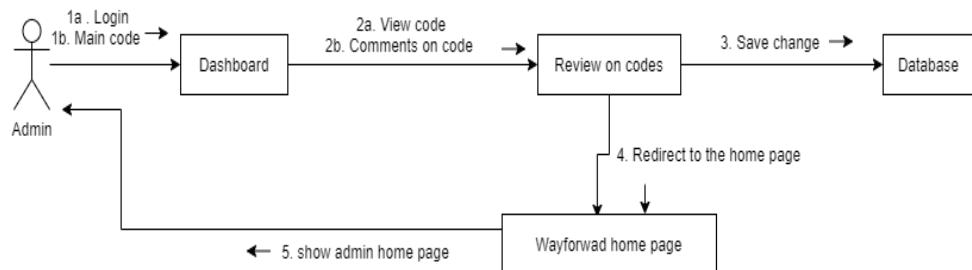


Figure 199 Collaboration Diagram for Review on Code

4.7.25 Ratings

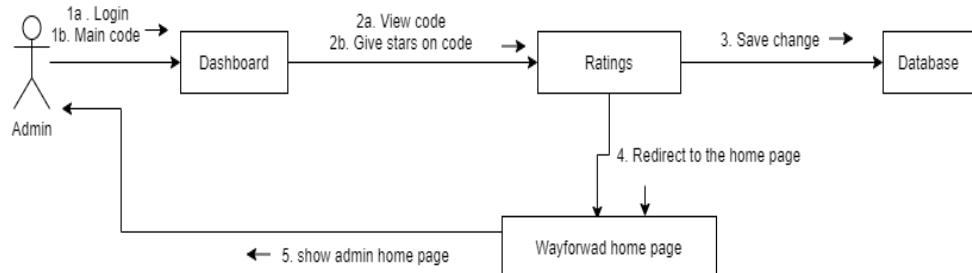


Figure 200 Collaboration Diagram for Ratings

4.7.26 Ask Query

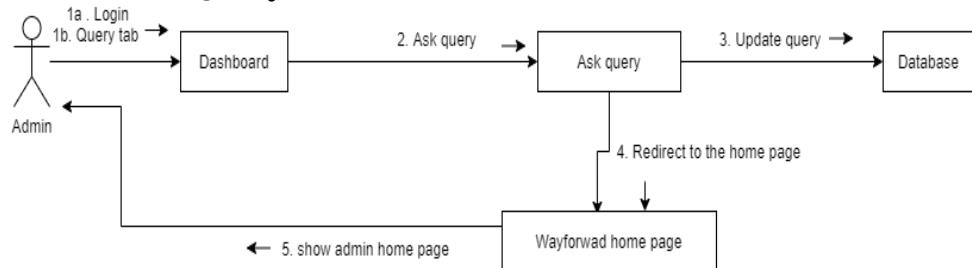


Figure 201 Collaboration Diagram for Ask Query

4.7.27 Solution of Query

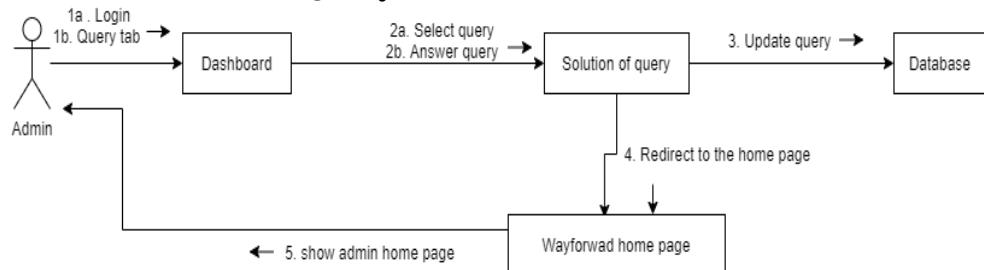


Figure 202 Collaboration Diagram for Solution of Query

4.7.28 Admin Upload Codes

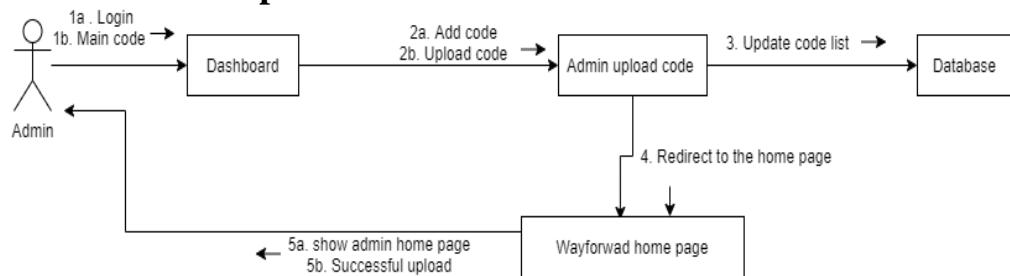


Figure 203 Collaboration Diagram for Admin Upload Codes

4.7.29 User Upload Codes

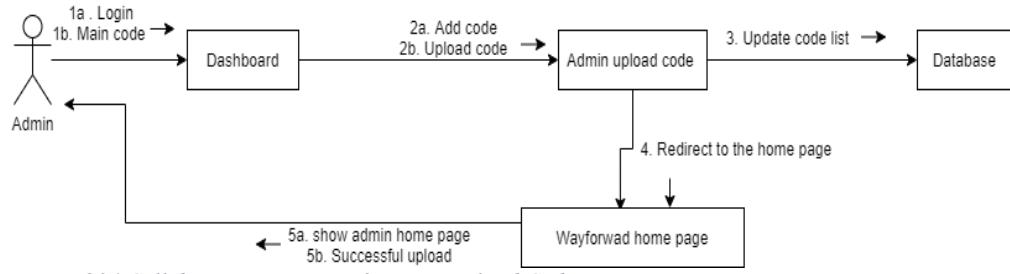


Figure 204 Collaboration Diagram for User Upload Codes

4.7.30 User Rate Script

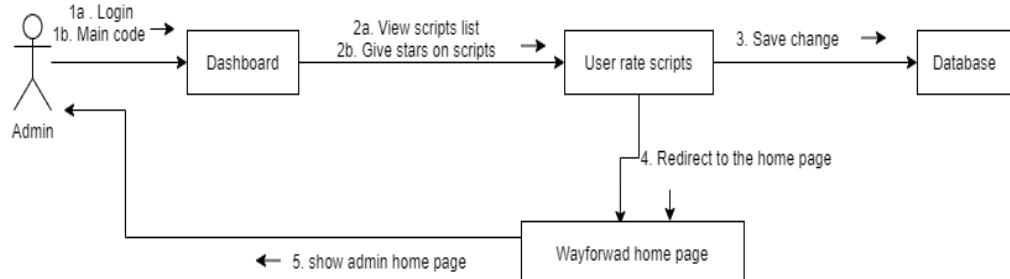


Figure 205 Collaboration Diagram for User Rate Script

4.7.31 Switch Roles

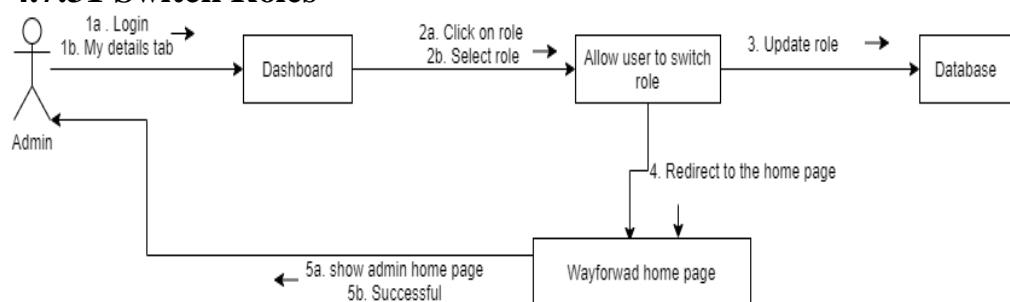


Figure 206 Collaboration Diagram for Switch Roles

4.7.32 User Participate in Voting

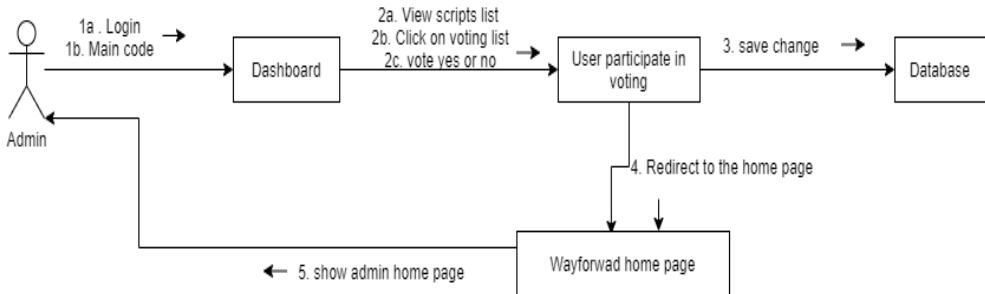


Figure 207 Collaboration Diagram for User Participate in Voting

4.7.33 Upload Script

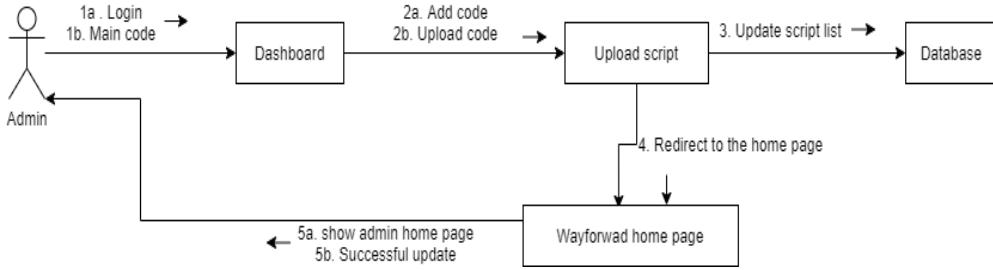


Figure 208 Collaboration Diagram for Upload Script

4.7.34 Delete Script

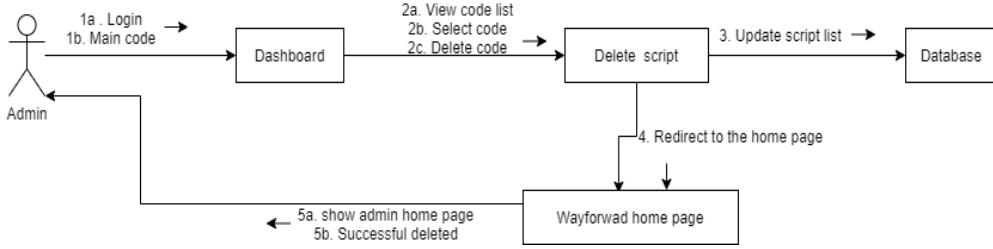


Figure 209 Collaboration Diagram for Delete Script

4.7.35 Update Script

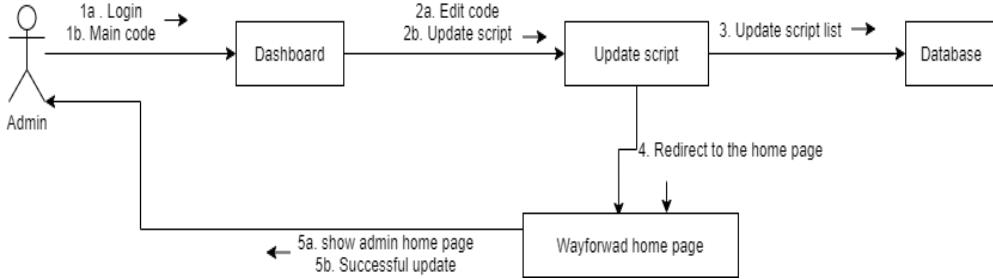


Figure 210 Collaboration Diagram for Update Script

4.7.36 User Ask Query

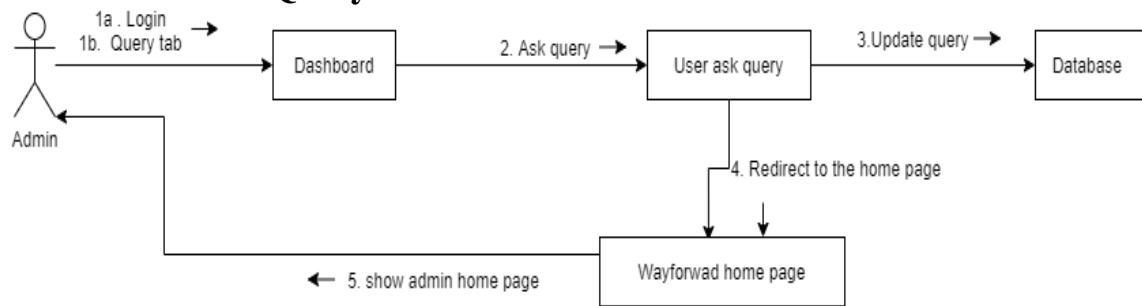


Figure 211 Collaboration Diagram for User Ask Query

4.8 State Transition Diagram

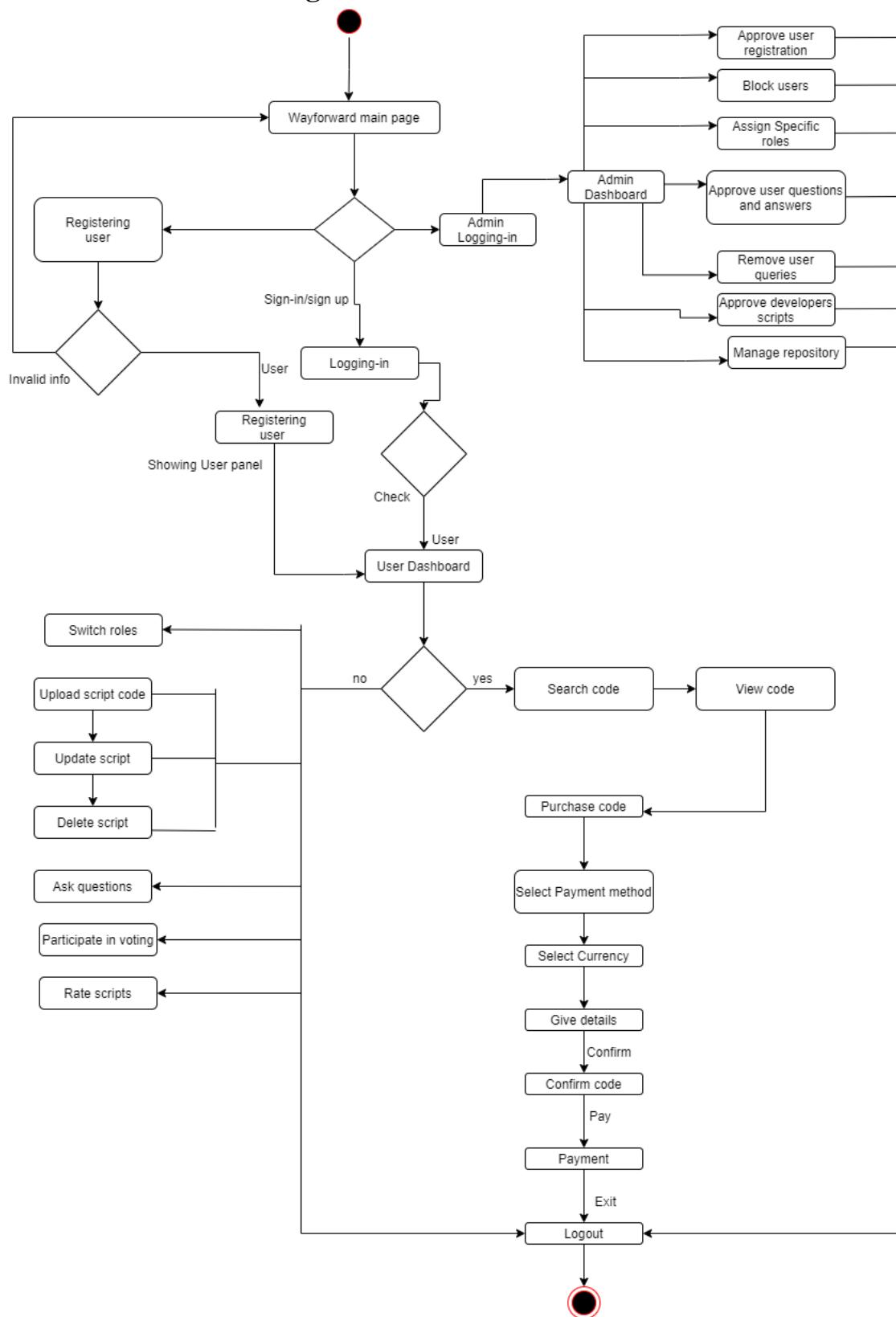


Figure 212 State Transition Diagram

4.9 Component Diagram

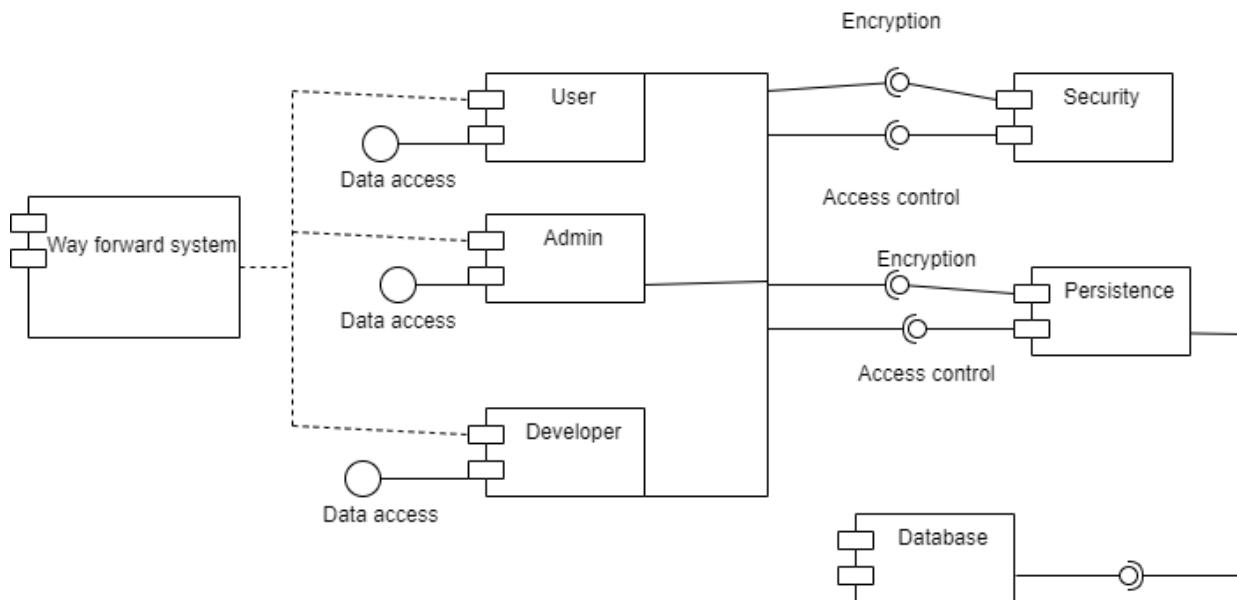


Figure 213 Component Diagram

4.10 Deployment Diagram

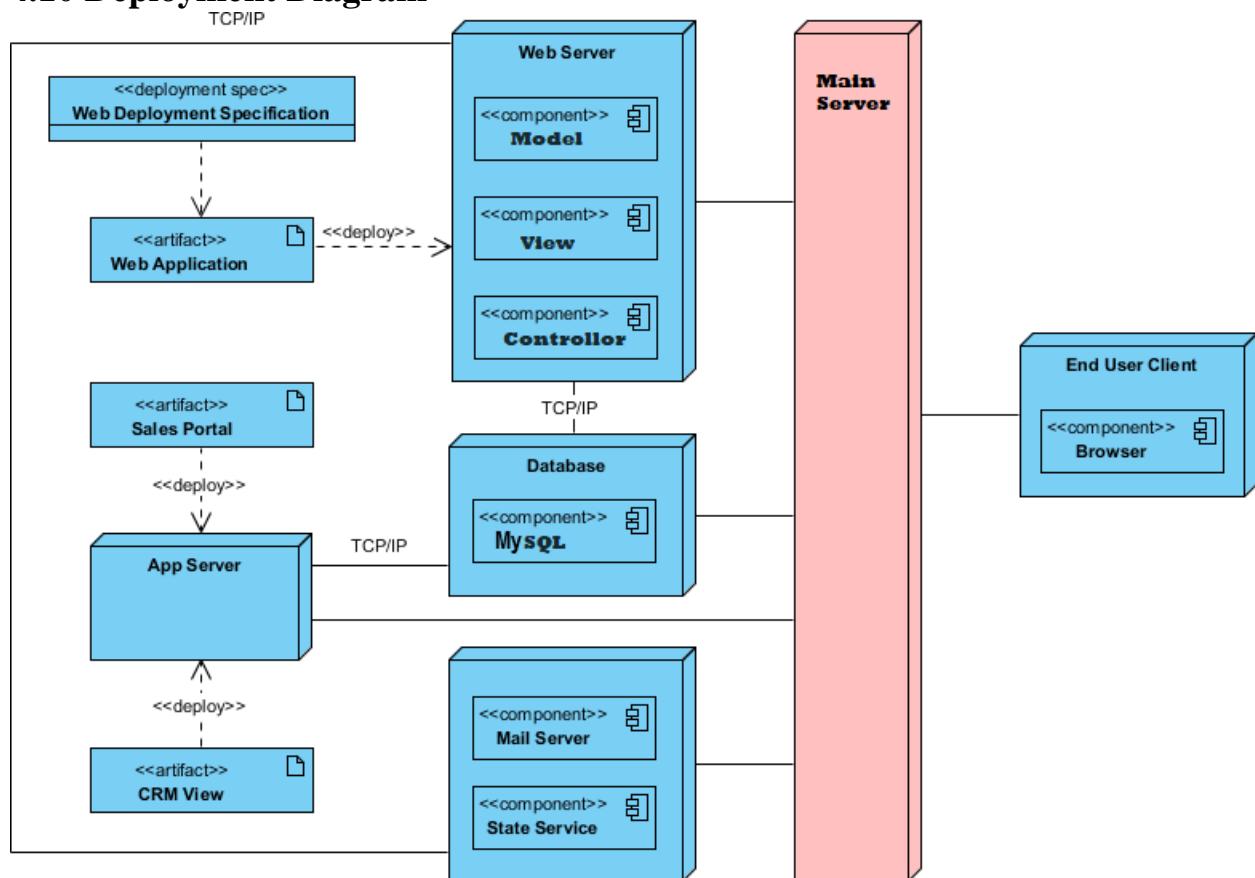


Figure 214 Deployment Diagram