DENTNUTZ

ปฤณสิษฐ์ จำปา (leader & scrum master) [Epic 1 story 1]	6633143321
ภัทรดนัย อาศรมเงิน [Epic 2 story 1]	6633186321
กษิดิศ นิตย์วิบูลย์ [Epic 2 story 1]	6633013421
ธีรุตม์ สุวรรณระดา [Epic 2 story 2]	6633115821
อชิตะ ฉลองศิริกุณ [Epic 1 story 2]	6633275521
ธนาธิป กัญญาบุญ [Epic 1 story 4]	6633103221
ณัฐพล ธีรยุทธวงศ์ [Epic 2 story 3]	6633072421
สิรธีร์ โตประดิษฐ์ [Epic 2 story 4]	6633260021
วรพล ศรีวิเศษ [Epic 1 story 3]	6633216521

4/30/2024

Epic 1

User Story 1 : As a Dentist I want to be able to view my appointments in a form of schedule so that I can effectively manage my schedule and patient's appointments

Equivalence classes

1.	Token isn’t null	Valid
2.	Token is null	Invalid
3.	User’s role = user, when Token isn’t null	Valid
4.	User’s role = admin, when Token isn’t null	Invalid
5.	User’s type = dentist, when Token isn’t null	Valid
6.	User’s type = patient, when Token isn’t null	Invalid

Test Case

Test Case ID: TC1
Input(s): User’s role = user
User’s type = dentist
Token isn’t null
Expected Output(s): Valid
Classes Covered: 1,3,5

Test Case ID: TC2
Input(s): User’s role = user
User’s type = patient
Token isn’t null
Expected Output(s): Invalid
Classes Covered: 1,3,6

Test Case ID: TC3
Input(s): User’s role = admin
Token isn’t null
Expected Output(s): Invalid
Classes Covered: 1,4

Test Case ID: TC4
Input(s): Token isn’t null
Expected Output(s): Invalid
Classes Covered: 2

User Story 2 : As a Admin I want to be able to access any patient appointment page from the schedule page so that I can access the appointment and edit it

Equivalence classes

1. Token isn’t null Valid
2. Token is null Invalid
3. User’s role = user, when Token isn’t null Invalid
4. User’s role = admin, when Token isn’t null Valid
5. User’s type = dentist, when Token isn’t null Invalid
6. User’s type = patient, when Token isn’t null Valid

Test Case

Test Case ID: TC1
Input(s): User’s role = user
User’s type = dentist
Token isn’t null
Expected Output(s): Invalid
Classes Covered: 1,3,5

Test Case ID: TC2
Input(s): User’s role = user
User’s type = patient
Token isn’t null
Expected Output(s): Invalid
Classes Covered: 1,3,6

Test Case ID: TC3
Input(s): User’s role = admin
Token isn’t null
Expected Output(s): Valid
Classes Covered: 1,4

Test Case ID: TC4
Input(s): Token isn’t null
Expected Output(s): Invalid
Classes Covered: 2

User Story 3 : As a Dentist I want to be able to access my patient's appointment from my schedule page so that I can access the appointment and create Report from it

Equivalence classes

1.	Token isn’t null	Valid
2.	Token is null	Invalid
3.	User’s role = user, when Token isn’t null	Valid
4.	User’s role = admin, when Token isn’t null	Invalid
5.	User’s type = dentist, when Token isn’t null	Valid
6.	User’s type = patient, when Token isn’t null	Invalid

Test Case

Test Case ID: TC1
Input(s): User’s role = user
User’s type = dentist
Token isn’t null
Expected Output(s): Valid
Classes Covered: 1,3,5

Test Case ID: TC2
Input(s): User’s role = user
User’s type = patient
Token isn’t null
Expected Output(s): Invalid
Classes Covered: 1,3,6

Test Case ID: TC3
Input(s): User’s role = admin
Token isn’t null
Expected Output(s): Invalid
Classes Covered: 1,4

Test Case ID: TC4
Input(s):   Token is null
Expected Output(s): Invalid
Classes Covered: 2

User Story 4 : As Admin I want to know how many appointment in each day so that I can manage number of dentists standby

Equivalence classes

1.	Token isn’t null	Valid
2.	Token is null	Invalid
3.	User’s role = user, when Token isn’t null	Invalid
4.	User’s role = admin, when Token isn’t null	Valid
5.	User’s type = dentist, when Token isn’t null	Invalid
6.	User’s type = patient, when Token isn’t null	Valid

Test Case

Test Case ID: TC1
Input(s): User’s role = user
User’s type = dentist
Token isn’t null
Expected Output(s): Invalid
Classes Covered: 1,3,5

Test Case ID: TC2
Input(s): User’s role = user
User’s type = patient
Token isn’t null
Expected Output(s): Invalid
Classes Covered: 1,3,6

Test Case ID: TC3
Input(s): User’s role = admin
Token isn’t null
Expected Output(s): Valid
Classes Covered: 1,4

Test Case ID: TC4
Input(s): Token is null
Expected Output(s): Invalid
Classes Covered: 2

Epic 2

User Story 1 : As a Dentist I want to be able to create patient's dental report so that for each appointment I can summarize my patient oral health's condition and report it to them

Equivalence classes

1.	Token isn’t null	Valid
2.	Token is null	Invalid
3.	User’s role = user, when Token isn’t null	Valid
4.	User’s role = admin, when Token isn’t null	Invalid
5.	User’s type = dentist, when Token isn’t null	Valid
6.	User’s type = patient, when Token isn’t null	Invalid
7.	patientId is mongoose.Schema.ObjectId	Valid
8.	patientId is null	Invalid
9.	appointmentId is mongoose.Schema.ObjectId	Valid
10.	appointmentId is null	Invalid
11.	dentistId is mongoose.Schema.ObjectId	Valid
12.	dentistId is null	Invalid
13.	treatment is string	Valid
14.	treatment is null	Invalid
15.	prescribed_medication is string	Valid
16.	prescribed_medication is null	Invalid
17.	recommendations is string	Valid
18.	recommendations is null	Invalid
19.	Date is from appointment which isn’t null	Valid
20.	Date is from appointment which is null	Invalid

Test Case

Test Case ID: TC1

Input(s): User’s role = user
User’s type = dentist
Token isn’t null
patientId is mongoose.Schema.ObjectId
appointmentId is mongoose.Schema.ObjectId
dentistId is mongoose.Schema.ObjectId
treatment is string
prescribed_medication is string
recommendations is string
Date is from appointment which isn’t null

Expected Output(s): Valid

Classes Covered: 1,3,5,7,9,11,13,15,17,19

Test Case ID: TC2

Input(s): User’s role = user
User’s type = patient
Token isn’t null
patientId is mongoose.Schema.ObjectId
appointmentId is mongoose.Schema.ObjectId
dentistId is mongoose.Schema.ObjectId
treatment is string
prescribed_medication is string
recommendations is string
Date is from appointment which isn’t null

Expected Output(s): Invalid

Classes Covered: 1,3,6,7,9,11,13,15,17,19

Test Case ID: TC3
Input(s): User’s role = admin
Token isn’t null
patientId is mongoose.Schema.ObjectId
appointmentId is mongoose.Schema.ObjectId
dentistId is mongoose.Schema.ObjectId
treatment is string
prescribed_medication is string
recommendations is string
Date is from appointment which isn’t null
Expected Output(s): Invalid
Classes Covered: 1,4,7,9,11,13,15,17,19

Test Case ID: TC4
Input(s): Token is null
patientId is mongoose.Schema.ObjectId
appointmentId is mongoose.Schema.ObjectId
dentistId is mongoose.Schema.ObjectId
treatment is string
prescribed_medication is string
recommendations is string
Date is from appointment which isn’t null
Expected Output(s): Invalid
Classes Covered: 2,7,9,11,13,15,17,19

Test Case ID: TC5

Input(s): User’s role = user
User’s type = dentist
Token isn’t null
patientId is null
appointmentId is mongoose.Schema.ObjectId
dentistId is mongoose.Schema.ObjectId
treatment is string
prescribed_medication is string
recommendations is string
Date is from appointment which isn’t null

Expected Output(s): Invalid

Classes Covered: 1,3,5,8,9,11,13,15,17,19

Test Case ID: TC6

Input(s): User’s role = user
User’s type = dentist
Token isn’t null
patientId is mongoose.Schema.ObjectId
appointmentId is null
dentistId is mongoose.Schema.ObjectId
treatment is string
prescribed_medication is string
recommendations is string
Date is from appointment which is null

Expected Output(s): Invalid

Classes Covered: 1,3,5,7,10,11,13,15,17,20

Test Case ID: TC7

Input(s): User’s role = user
User’s type = dentist
Token isn’t null
patientId is mongoose.Schema.ObjectId
appointmentId is mongoose.Schema.ObjectId
dentistId is null
treatment is string
prescribed_medication is string
recommendations is string
Date is from appointment which isn’t null

Expected Output(s): Invalid

Classes Covered: 1,3,5,7,9,12,13,15,17,19

Test Case ID: TC8

Input(s): User’s role = user
User’s type = dentist
Token isn’t null
patientId is mongoose.Schema.ObjectId
appointmentId is mongoose.Schema.ObjectId
dentistId is mongoose.Schema.ObjectId
treatment is null
prescribed_medication is string
recommendations is string
Date is from appointment which isn’t null

Expected Output(s): Invalid

Classes Covered: 1,3,5,7,9,11,14,15,17,19

Test Case ID: TC9

Input(s): User’s role = user
User’s type = dentist
Token isn’t null
patientId is mongoose.Schema.ObjectId
appointmentId is mongoose.Schema.ObjectId
dentistId is mongoose.Schema.ObjectId
treatment is string
prescribed_medication is null
recommendations is string
Date is from appointment which isn’t null

Expected Output(s): Invalid

Classes Covered: 1,3,5,7,9,11,13,16,17,19

Test Case ID: TC10

Input(s): User’s role = user
User’s type = dentist
Token isn’t null
patientId is mongoose.Schema.ObjectId
appointmentId is mongoose.Schema.ObjectId
dentistId is mongoose.Schema.ObjectId
treatment is string
prescribed_medication is string
recommendations is null
Date is from appointment which isn’t null

Expected Output(s): Invalid

Classes Covered: 1,3,5,7,9,11,13,15,18,19

User Story 2 : As a Patient I want to see my dental report so that I will know my oral health and treatment after each appointment

Equivalence classes

1.	Token isn’t null	Valid
2.	Token is null	Invalid
3.	User’s role = user, when Token isn’t null	Valid
4.	User’s role = admin, when Token isn’t null	Invalid
5.	User’s type = dentist, when Token isn’t null	Invalid
6.	User’s type = patient, when Token isn’t null	Valid

Test Case

Test Case ID: TC1
Input(s): User’s role = user
User’s type = patient
Token isn’t null
Expected Output(s): Valid
Classes Covered: 1,3,6

Test Case ID: TC2
Input(s): User’s role = user
User’s type = dentist
Token isn’t null
Expected Output(s): Invalid
Classes Covered: 1,3,5

Test Case ID: TC3
Input(s): User’s role = admin
User’s type = patient
Token isn’t null
Expected Output(s): Invalid
Classes Covered: 1,4,6

Test Case ID: TC4
Input(s):   User’s role = admin
               User’s type = dentist
               Token isn’t null
Expected Output(s): Invalid
Classes Covered: 1,4,5

Test Case ID: TC5
Input(s):   Token is null
Expected Output(s): Invalid
Classes Covered: 2

User Story 3 : As a Dentist I want to be able to see my patient's dental report so that I can monitor their treatment progress briefly

Equivalence classes

1.	Token isn’t null	Valid
2.	Token is null	Invalid
3.	User’s role = user, when Token isn’t null	Valid
4.	User’s role = admin, when Token isn’t null	Invalid
5.	User’s type = dentist, when Token isn’t null	Valid
6.	User’s type = patient, when Token isn’t null	Invalid

Test Case

Test Case ID: TC1
Input(s): User’s role = user
User’s type = patient
Token isn’t null
Expected Output(s): Valid
Classes Covered: 1,3,6

Test Case ID: TC2
Input(s): User’s role = user
User’s type = dentist
Token isn’t null
Expected Output(s): Invalid
Classes Covered: 1,3,5

Test Case ID: TC3
Input(s): User’s role = admin
User’s type = patient
Token isn’t null
Expected Output(s): Invalid
Classes Covered: 1,4,6

Test Case ID: TC4
Input(s):	User’s role = admin
	User’s type = dentist
	Token isn’t null
Expected Output(s): Invalid
Classes Covered: 1,4,5

Test Case ID: TC5
Input(s):	Token is null
Expected Output(s): Invalid
Classes Covered: 2

User Story 4 : As a Dentist I want to be able to update my patient's dental report so that I can change report's information if the information was wrong

Equivalence classes

1.	Token isn’t null	Valid
2.	Token is null	Invalid
3.	User’s role = user, when Token isn’t null	Valid
4.	User’s role = admin, when Token isn’t null	Invalid
5.	User’s type = dentist, when Token isn’t null	Valid
6.	User’s type = patient, when Token isn’t null	Invalid
7.	treatment is string	Valid
8.	treatment is null	Invalid
9.	prescribed_medication is string	Valid
10.	prescribed_medication is null	Invalid
11.	recommendations is string	Valid
12.	recommendations is null	Invalid

Test Case

Test Case ID: TC1
Input(s): User’s role = user
User’s type = dentist
Token isn’t null
Expected Output(s): Valid
Classes Covered: 1,3,5

Test Case ID: TC2
Input(s): User's role = user
User's type = patient
Token isn’t null
Expected Output(s): Invalid
Classes Covered: 1,3,6

Test Case ID: TC3
Input(s): User's role = user
User's type = dentist
Token isn’t null
treatment is null
prescribed_medication is null
recommendations is null
Expected Output(s): Invalid
Classes Covered: 1,3,5,8,10,12

Test Case ID: TC4
Input(s): User's role = admin
Token isn’t null
Expected Output(s): Invalid
Classes Covered: 1,4

Test Case ID: TC5
Input(s): User's role = user
User's type = dentist
Token isn’t null
treatment is string
prescribed_medication is null
recommendations is null
Expected Output(s): Invalid
Classes Covered: 1,3,5,7,10,12

Test Case ID: TC6
Input(s): User's role = user
User's type = patient
Token isn’t null
treatment is string
prescribed_medication is string
recommendations is string
Expected Output(s): Invalid
Classes Covered:1,3,6,7,9,11

Test Case ID: TC7
Input(s): Token is null
Expected Output(s): Invalid
Classes Covered: 2

User Story 5 : As a Admin I want to delete report that has resided within the system for an extended duration. so that the data will not be overflowed

*No input, Only set schema expire date