DriveIT

Korbela-Pokryvka-Stefkovic

Table of Contents

Stereotypes	4
Use Case Diagram	
Analysis Class Diagram	
GetExamResults	
LogIntoTheoretical	18
Application	20
ERD	
Design Class Diagram	24
CreatePracticalLesson	
ChoosePracticalLesson	28
ViewCourses	30
Communication Diagram Get Exam Dates	32
Communication Diagram View Courses	

Table of Figures

Use Case Diagram	6
Analysis Class Diagram	
GetExamResults	16
LogIntoTheoretical	18
Application	20
ERD	22
Design Class Diagram	24
CreatePracticalLesson	27
ChoosePracticalLesson	28
ViewCourses	30
Communication Diagram Get Exam Dates	32
Communication Diagram View Courses	

Stereotypes

UseCase				
Tagged Value Definitions	Level	Level		
	Туре	Enumeration		
	Enumeration Values	Summary, User, Subfunction		
	Complexity			
	Туре	Enumeration		
	Enumeration Values	Low, Medium, High		
	Use Case Status			
	Туре	Enumeration		
	Enumeration Values	Name Only, Initial, Base, Complete, Deferred		
	Implementation Status	Implementation Status		
	Туре	Enumeration		
	Enumeration Values	Scheduled, Started, Partially Complete, Complete, Partially Deferred		
	Preconditions			
	Туре	Multi-line Text		
	Post-conditions			
	Туре	Multi-line Text		
	Author			
	Туре	Text		
	Assumptions			
	Туре	Multi-line Text		

Case Story

	·		
requirement			
Tagged Value Definitions	Text		
	Туре	HTML	
	ID		
	Туре	Text	
	source		
	Туре	Text	
	kind		
	Туре	Enumeration	
	Enumeration Values	Functional, Performance, Interface	
	verifyMethod		
	Туре	Enumeration	
	Enumeration Values	Analysis, Demonstration, Inspection, Test	
	risk		
	Туре	Enumeration	
	Enumeration Values	High, Medium, Low	
	status		
	Туре	Enumeration	
	Enumeration Values	Proposed, Approved, Rejected, Deferred, Implemented, Mandatory, Obsolete	
Include			
Extend			
use			
ORM Abstract Persistable			

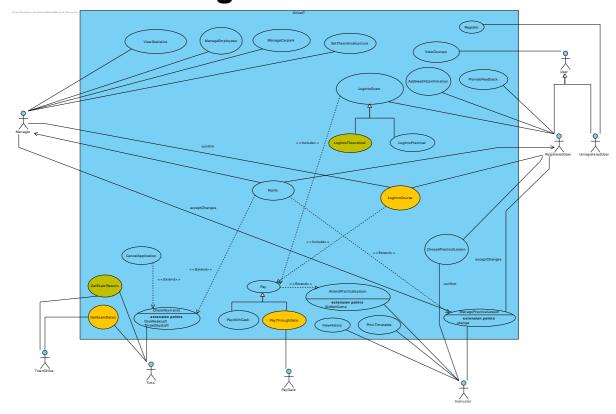
control

entity

boundary

Use Case Diagram

Use Case Diagram



Summary

Name	Description
♀ Manager	
₹ TownOffice	
₹ Time	
♀ PayGate	
₹ Instructor	
₹ RegisteredUser	
रू User	
♀ UnregisteredUser	
GetExamResults	
ViewStatistics	
ManageEmployee	3
ManageCarpark	
ViewHistory	
PrintTimetable	
ManagePracticalLe	esson
SetTheoreticalLec	ure
GetExamDates	Main Flow of Events:

	The use case starts when current time reached one week from last event. System checks if exam dates have been published by Town Office. IF exam dates are accessible System downloads exam dates to its database. System sets exam dates in database as available. Alternative Flows: If exam dates were not accessible, Manager is informed and Time starts
	counting another one week period.
LogIntoPractical	
LogIntoTheoretica	
LogIntoExam	
ChoosePracticalLe	
AttendPracticalLes	son
ProvideFeedback	
AddHealthConfirm	ation
CancelApplication	
Notify	
CheckPayments	
■ PayThroughGate	Main Flow of Events: 1. The use case starts when Pay is invoked by another use case and the gate option is chosen. 2. RegisteredUser is redirected to the PayGate. 3. RegisteredUser is asked to provide his/her credentials. 4. IF the credentials are recognized and correct 4.1 RegisteredUser is presented with a form of payment to fill in. 4.2 IF RegisteredUser chooses "Pay", 4.2.1 PayGate tries to proceed with the payment. 4.2.2 IF the payment was successful 4.2.2.1 RegisteredUser is notified about successful operation and money is transferred. 4.2.3 ELSE 4.2.3.1 RegisteredUser is notified about a problem. 4.3 ELSE IF RegisteredUser chooses "Cancel" 4.3.1 Operation is canceled. Alternative flows: If the credentials were not recognized or incorrect, RegisteredUser is prompted again.
PayWithCash	
Pay	
■ LogIntoCourse	Main Flow of Events: 1. The use case starts when a RegisteredUser chooses "Log Into Course". 2. System returns lists of available courses. 3. RegisteredUser chooses a course. 4. IF RegisteredUser meets course preconditions 4.1. System displays price and other details. 4.2. INCLUDE Pay

	4.3. IF Payment was successful4.3.1. Send application to Manager for approval.4.3.2. IF Manager approves4.3.2.1. Assign RegisteredUser to course.
	Alternate Flows: If RegisteredUser does not meet course preconditions, system display message and returns to previous menu. If Payment was not successful three days before course starts, application is canceled. If Manager does not approve, application is canceled.
Register	
ViewCourses	
DriveIT	

- ₹ Manager
- **₹ TownOffice**
- ₹ Time
- ₹ PayGate
- **₹ Instructor**
- ₹ RegisteredUser
- [₹] User
- **₹ UnregisteredUser**
- ⊕EN/A
- **₩**EN/A
- **₩**EN/A
- GetExamResults
- ViewStatistics
- ManageEmployees
- ManageCarpark

- ViewHistory
- PrintTimetable
- ManagePracticalLesson

Extension Points

change

- SetTheoreticalLecture
- GetExamDates

Details

Name	Value
Preconditions	One week period has passed. Exam dates have been published by Town Office.
Post-conditions	Exam dates are inserted into system.

- LogIntoPractical
- LogIntoTheoretical
- LogIntoExam
- ChoosePracticalLesson
- AttendPracticalLesson

Extension Points

DidNotCome

- ProvideFeedback
- F. N/A
- AddHealthConfirmation
- CancelApplication

Notify

CheckPayments

Extension Points

OneWeekLeft

ThreeDaysLeft

PayThroughGate

Details

Name	Value
Preconditions	RegisteredUser is client of online banking.
Post-conditions	none

PayWithCash

Pay

LogIntoCourse

Details

Name	Value
Preconditions	RegisteredUser is logged in the system.
Post-conditions	RegisteredUser is enrolled in a course.

Register

ViewCourses

DriveIT

Children

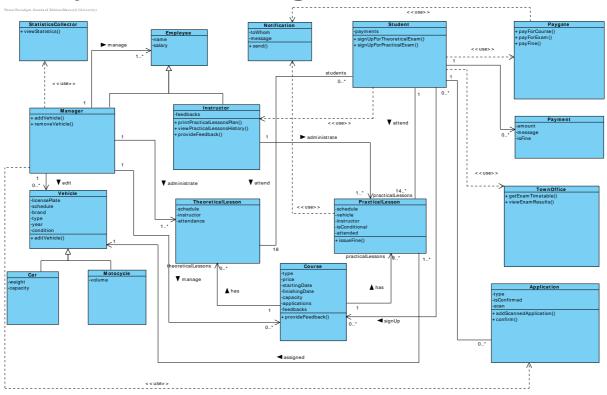
Name	Description
ViewCourses	
Register	
LogIntoCourse	Main Flow of Events: 1. The use case starts when a RegisteredUser chooses "Log Into Course". 2. System returns lists of available courses. 3. RegisteredUser chooses a course. 4. IF RegisteredUser meets course preconditions 4.1. System displays price and other details. 4.2. INCLUDE Pay 4.3. IF Payment was successful

	4.3.1. Send application to Manager for approval.4.3.2. IF Manager approves4.3.2.1. Assign RegisteredUser to course.
	Alternate Flows: If RegisteredUser does not meet course preconditions, system display message and returns to previous menu. If Payment was not successful three days before course starts, application is canceled. If Manager does not approve, application is canceled.
Pay	
PayWithCash	
■ PayThroughGate	Main Flow of Events: 1. The use case starts when Pay is invoked by another use case and the gate option is chosen. 2. RegisteredUser is redirected to the PayGate. 3. RegisteredUser is asked to provide his/her credentials. 4. IF the credentials are recognized and correct 4.1 RegisteredUser is presented with a form of payment to fill in. 4.2 IF RegisteredUser chooses "Pay", 4.2.1 PayGate tries to proceed with the payment. 4.2.2 IF the payment was successful 4.2.2.1 RegisteredUser is notified about successful operation and money is transferred. 4.2.3 ELSE 4.2.3.1 RegisteredUser is notified about a problem. 4.3 ELSE IF RegisteredUser chooses "Cancel" 4.3.1 Operation is canceled. Alternative flows: If the credentials were not recognized or incorrect, RegisteredUser is prompted again.
CheckPayments	internett, registeredeser is prompted again.
Notify	
CancelApplication	
AddHealthConfirmation	
ProvideFeedback	
AttendPracticalLesson	
ChoosePracticalLesson	
LogIntoExam	
LogIntoTheoretical	
LogIntoPractical	
GetExamDates	Main Flow of Events: 1. The use case starts when current time reached one week from last event. 2. System checks if exam dates have been published by Town Office.

	 3. IF exam dates are accessible 3.1 System downloads exam dates to its database. 3.2 System sets exam dates in database as available. Alternative Flows: If exam dates were not accessible, Manager is informed and Time starts counting another one week period.
SetTheoreticalLecture	
ManagePracticalLesson	
PrintTimetable	
ViewHistory	
ManageCarpark	
ManageEmployees	
ViewStatistics	
GetExamResults	

Class Diagram

Analysis Class Diagram



Summary

Name	Description
StatisticsCollector	
Student	
Paygate	
Notification	
Employee	
Instructor	
Manager	
Payment	
TownOffice	
Vehicle	
TheoreticalLesson	
PracticalLesson	
Course	
Motocycle	
Car	
Application	

Description

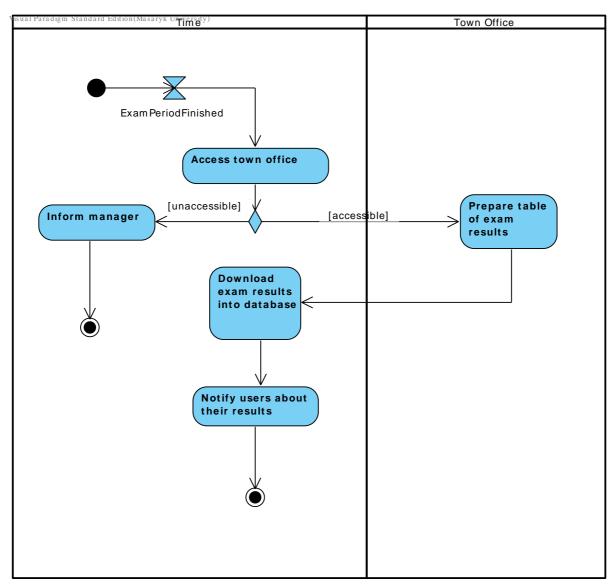
DriveIT Analysis Class Diagram

- StatisticsCollector
- Student
- Paygate
- Notification
- Employee
- ⊕EN/A
- Instructor
- Manager
- Payment
- **TownOffice**
- Vehicle
- TheoreticalLesson

- PracticalLesson
- ⊕EN/A
- Course
- **■** Motocycle
- Car
- Application

Activity Diagram

GetExamResults



Summary

Name	Description
Swimlane	
Time	
Town Office	
ExamPeriodFinished ■	
Access town office	
Inform manager	
Prepare table of exam results	
Download exam results into	
database	
Notify users about their results	

Swimlane Activity Partitions

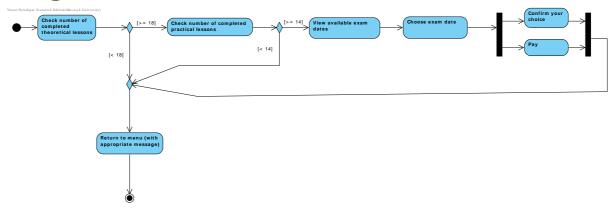
Time		
Contained Elements	Name	Description
	Access town office	
	N/A	
	→<ٍ N/A	
	Inform manager	
	● N/A	
	Download exam results into	
	database	
	Notify users about their	
	results	
	● N/A	
	ExamPeriodFinished	

Town Office		
Contained Elements	Name	Description
	Prepare table of exam	
	results	

- N/A
- ExamPeriodFinished
- Access town office
- **→** N/A
- Inform manager
- N/A
- Prepare table of exam results
- Download exam results into database
- Notify users about their results
- N/A

Activity Diagram

LogIntoTheoretical



Summary

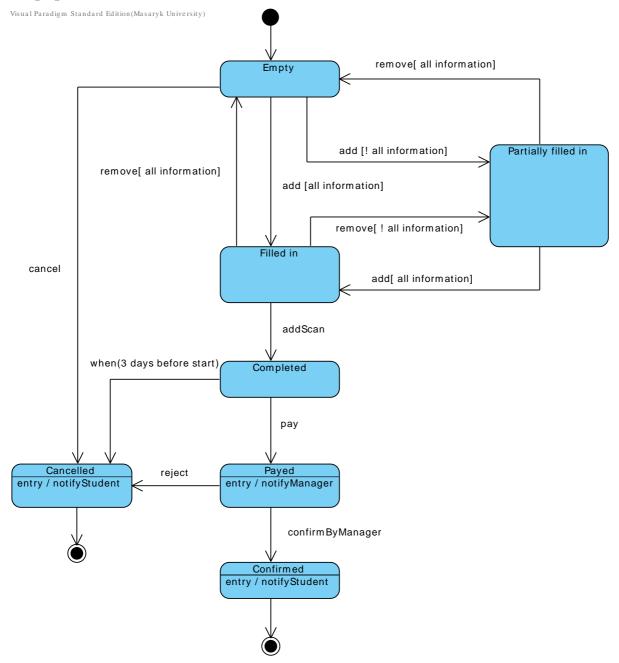
Name	Description
Check number of completed	
theoretical lessons	
Check number of completed	
practical lessons	
View available exam dates	
Choose exam date	
Confirm your choice	
Pay	
Return to menu (with appropriate	
message)	

- N/A
- Check number of completed theoretical lessons
- →N/A
- Check number of completed practical lessons
- **→**\$\\A
- View available exam dates
- Choose exam date

- ∜N/A
- Confirm your choice
- Pay
- ≯N/A
- **→** N/A
- Return to menu (with appropriate message)
- ●N/A

State Machine Diagram

Application



Summary

Name	Description
Empty	
Filled in	
Completed	
Payed	
Confirmed	
Cancelled	

Partially filled in	

- N/A
- Empty
- Filled in
- Completed
- Payed

Regions

Name	Description
notifyManager	

Confirmed

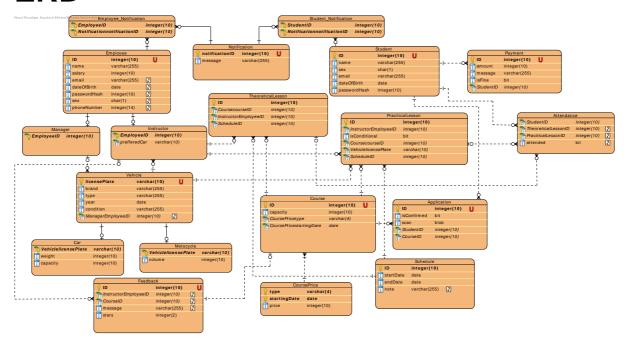
Regions

Name	Description
notifyStudent	

- N/A
- Cancelled
- N/A
- Partially filled in

Entity Relationship Diagram

ERD



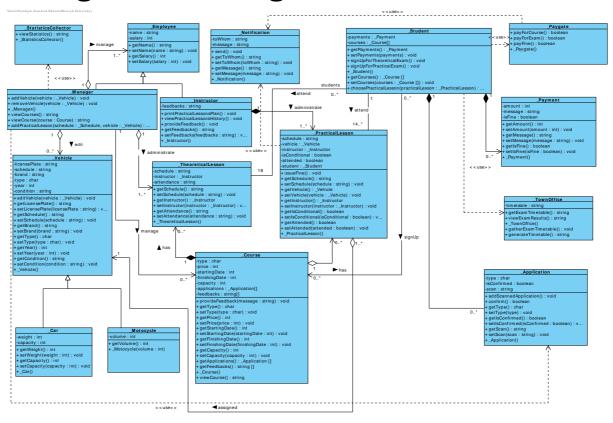
Summary

Name	Description.
Name	Description
Employee_Notification	tion
Student_Notification	n
Notification	
Student	
Employee	
Payment	
TheoreticalLesson	
Attendance	
PraciticalLesson	
Instructor	
Manager	
Vehicle	
Course	
Application	
Car	
Motocycle	
Schedule	
Feedback	
CoursePrice	

- Employee_Notification
- Student_Notification
- Notification
- Student
- Employee
- Payment
- TheoreticalLesson
- Attendance
- PraciticalLesson
- Instructor
- Manager
- Vehicle
- Course
- Application
- Car
- Motocycle
- Schedule
- Feedback
- CoursePrice

Class Diagram

Design Class Diagram



Summary

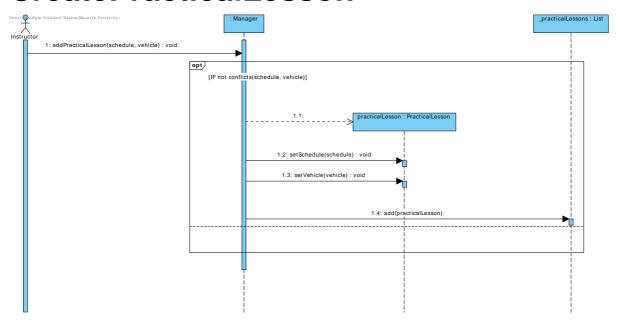
Name	Description
Employee	
Paygate	
StatisticsCollecto	r
Student	
Notification	
Manager	
Payment	
_Instructor	
PracticalLesson	
Vehicle	
_TheoreticalLesso	n
TownOffice	
Course	
Application	
Car	
Motocycle	

- _Employee
- Paygate
- StatisticsCollector
- Student
- Notification
- ⊕EN/A
- _Manager
- Payment
- _Instructor
- PracticalLesson
- Vehicle
- TheoreticalLesson
- _TownOffice

- _Course
- _Application
- ⊕N/A
- Car
- _Motocycle

Sequence Diagram

CreatePracticalLesson



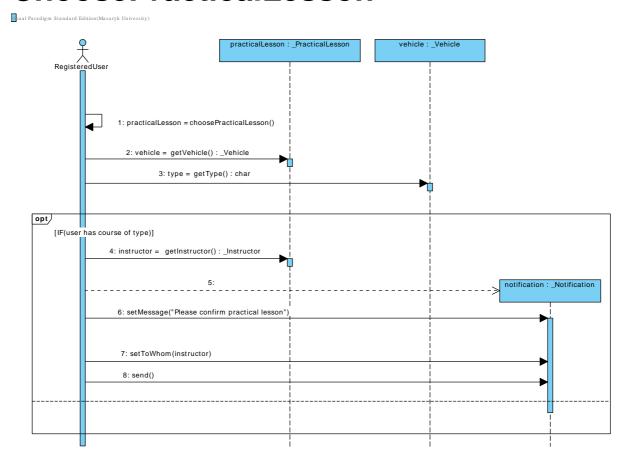
Summary

Name	Description
웃 Instructor	
: Manager	
practicalLesson : PracticalLesson	
_practicalLessons : List	
IF not conflicts	
Operand2	
□ alt	

- **₹ Instructor**
- 🕇 : Manager
- TpracticalLesson: PracticalLesson
- T_practicalLessons : List
- **■IF** not conflicts
- Operand2
- **alt**

Sequence Diagram

ChoosePracticalLesson



Summary

Name	Description
웃 RegisteredUser	
practicalLesson : _PracticalLesson	
vehicle : _Vehicle	
notification : _Notification	
Operand4	
Operand5	
IF(user has course of type)	

Details

₹ RegisteredUser

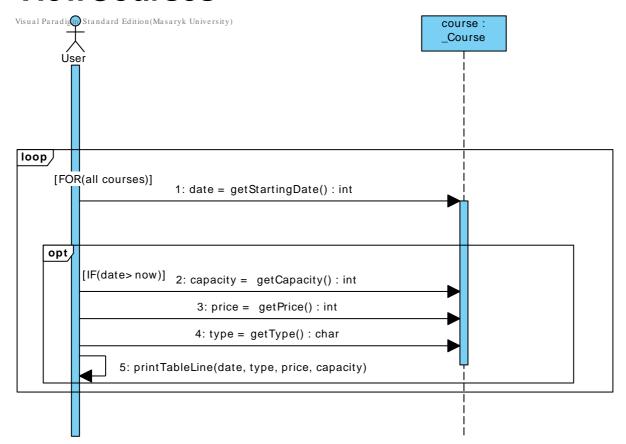
TpracticalLesson: _PracticalLesson

vehicle : _Vehicle

- notification: _Notification
- Operand4
- Operand5
- IF(user has course of type)

Sequence Diagram

ViewCourses



Summary

Name	Description
₹ User	
course : _Course	
Operand	
Operand3	
IF(date>now)	
FOR(all courses)	

Details

[₹] User

course : _Course

Operand

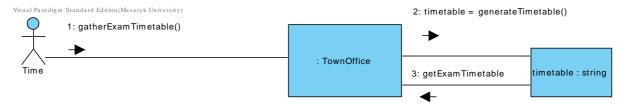
Children

Name	Description
IF(date>now)	

- Operand3
- IF(date>now)
- FOR(all courses)

Communication Diagram

Communication Diagram Get Exam Dates



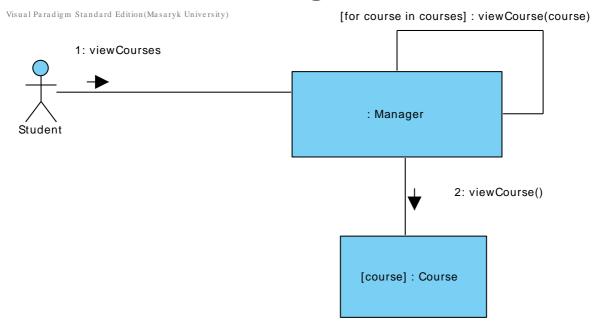
Summary

Name	Description
웃 Time	
⇒ gatherExamTimetable()	
: TownOffice	
⇒ timetable = generateTimetable()	
timetable : string	
→ getExamTimetable	

- [₹] Time
- →gatherExamTimetable()
- **T**: TownOffice
- →timetable = generateTimetable()
- Ttimetable: string
- → getExamTimetable

Communication Diagram

Communication Diagram View Courses



Summary

Name	Description
♀ Student	
→ viewCourses	
: Manager	
→ viewCourse()	
[course] : Course	

Details

? Student

→viewCourses

T: Manager

→viewCourse()

T [course] : Course