

# CSE 343 – Project Final Report

## Project: Yan Koltuk

By ZetshSoft

---

**Scrum Master:** Talha Kaymak (200104004052)

**Product Owner:** Ahmet Hakan Demir (210104004055)

**Developers:**

- Sümeyye Zeynep Gürbüz (200104004039)
  - Ebrar Özkan (220104004005)
  - Hamide Sıla Akdan (220104004063)
-

## User Requirements Document

OVERVIEW	This document defines the requirements of the service tracking system to be developed for schools. The system aims to make the process of students getting on and off school buses traceable by families. It will be designed to be usable on both mobile and web-based platforms.
GOAL	To ensure that students registered in school services can be monitored in real time by their families. Thus, the safety of students will be increased, and service processes will be made more transparent.
TARGET GROUP	<ul style="list-style-type: none"><li>– Families of primary and secondary school students</li><li>– School bus drivers and stewardesses</li><li>– School bus drivers' administration</li></ul>

## User Roles and Features

ADMIN (Web)	<ul style="list-style-type: none"><li>– <b>Login:</b> Login via the web with the username and the password. (Admin is the owners of the Yan Koltuk.)</li><li>– <b>Manager Management:</b><ul style="list-style-type: none"><li>○ Manager registration:<ul style="list-style-type: none"><li>▪ Username</li></ul></li><li>○ Delete service</li></ul></li><li>– <b>Manager List:</b><ul style="list-style-type: none"><li>○ Can view the registered managers list.</li></ul></li><li>– <b>Interface:</b><ul style="list-style-type: none"><li>○ Navbar<ul style="list-style-type: none"><li>▪ A logout button</li><li>▪ Username is displayed</li></ul></li><li>○ 2 main buttons on the web page:<ul style="list-style-type: none"><li>▪ Add Manager</li><li>▪ View Manager List</li></ul></li></ul></li></ul>
-------------	--

	<ul style="list-style-type: none"> <li>○ Footer</li> </ul>
MANAGER (Web)	<ul style="list-style-type: none"> <li>– <b>Login:</b> Login via the web with the ID and the password given when applying for the product.</li> <li>– <b>Service Management:</b> <ul style="list-style-type: none"> <li>○ Service registration: <ul style="list-style-type: none"> <li>▪ License plate</li> <li>▪ Capacity</li> <li>▪ Point of departure</li> <li>▪ Departure time</li> <li>▪ Driver's name-surname</li> <li>▪ Driver's phone number</li> <li>▪ Driver's Turkish Id number</li> <li>▪ Stewardess' name-surname</li> <li>▪ Stewardess' phone number</li> <li>▪ Stewardess' Turkish Id number</li> </ul> </li> <li>○ Edit service data: <ul style="list-style-type: none"> <li>▪ Driver's name-surname</li> <li>▪ Driver's phone number</li> <li>▪ Driver's Turkish Id number</li> <li>▪ Stewardess' name-surname</li> <li>▪ Stewardess' phone number</li> <li>▪ Stewardess' Turkish Id number</li> </ul> </li> <li>○ Delete service</li> </ul> </li> <li>– <b>Service List:</b> <ul style="list-style-type: none"> <li>○ Can view the registered service list.</li> <li>○ When a service is clicked, it will show the service data, and the student list registered in that service.</li> </ul> </li> <li>– <b>Reporting:</b> <ul style="list-style-type: none"> <li>○ Can view the registered service list.</li> <li>○ When a service is clicked, an excel file will be downloaded showing the movements of the selected service.</li> </ul> </li> <li>– <b>Interface:</b> <ul style="list-style-type: none"> <li>○ Navbar <ul style="list-style-type: none"> <li>▪ A logout button</li> <li>▪ Username is displayed</li> </ul> </li> </ul> </li> </ul>

	<ul style="list-style-type: none"> <li>○ 3 main buttons on the web page: <ul style="list-style-type: none"> <li>▪ Add Service</li> <li>▪ View Service List</li> <li>▪ Service Reports</li> </ul> </li> <li>○ Footer</li> </ul>
PARENT (Mobile)	<ul style="list-style-type: none"> <li>– <b>Registration and Login:</b> <ul style="list-style-type: none"> <li>○ Registers with: <ul style="list-style-type: none"> <li>▪ Phone number</li> <li>▪ Password</li> <li>▪ Name-surname</li> <li>▪ Address</li> <li>▪ Turkish Id number</li> </ul> </li> <li>○ Log into the system with the registered phone number and the password.</li> </ul> </li> <li>– <b>Home Page:</b> <ul style="list-style-type: none"> <li>○ Includes a bottom navbar to: <ul style="list-style-type: none"> <li>▪ Add a student with: <ul style="list-style-type: none"> <li>• Name-surname</li> <li>• School number</li> <li>• Turkish Id number</li> <li>• Service plate</li> <li>• Pick-up location of the student</li> </ul> </li> <li>▪ View the parent's profile</li> <li>▪ Edit the parent's info: <ul style="list-style-type: none"> <li>• Phone number</li> <li>• Address</li> </ul> </li> </ul> </li> </ul> </li> <li>– <b>Student Tracking:</b> <ul style="list-style-type: none"> <li>○ The registered students are listed with these data: <ul style="list-style-type: none"> <li>▪ Name-surname</li> <li>▪ Registered service's plate</li> <li>▪ Get-in / get-off status</li> </ul> </li> </ul> </li> <li>– <b>Student management:</b> <ul style="list-style-type: none"> <li>○ A parent can request leave from the system for the days their child will not attend.</li> <li>○ Can delete the student from the system.</li> <li>○ Can change the service information of a student by changing the student's service plate.</li> </ul> </li> </ul>

	<ul style="list-style-type: none"> <li>– <b>Interface:</b> <ul style="list-style-type: none"> <li>○ User-friendly, simple navigation.</li> <li>○ Ease of adding and managing students.</li> </ul> </li> </ul>
SERVICE DRIVER (Mobile)	<ul style="list-style-type: none"> <li>– <b>Login:</b> Log into the system with the service plate and automatically generated password.</li> <li>– <b>Home Page:</b> <ul style="list-style-type: none"> <li>○ Includes a navbar for logout button.</li> <li>○ Includes a bottom navbar to: <ul style="list-style-type: none"> <li>▪ View and manage the registered students: <ul style="list-style-type: none"> <li>• Students are listed with these data: <ul style="list-style-type: none"> <li>○ Name-surname</li> <li>○ School number</li> <li>○ Parent phone number</li> </ul> </li> <li>• Can add note for each student</li> <li>• Can sort and organize the students' order</li> <li>• When a student is clicked, student's details will be displayed</li> </ul> </li> </ul> </li> <li>○ Displays the service data such as: <ul style="list-style-type: none"> <li>▪ Service details</li> <li>▪ Driver's details</li> <li>▪ Stewardess' details</li> </ul> </li> <li>○ Includes 2 main buttons for taking attendance <ul style="list-style-type: none"> <li>▪ Go to school</li> <li>▪ Return from school</li> </ul> </li> </ul> </li> <li>– <b>Student status update:</b> <ul style="list-style-type: none"> <li>○ Each student has a "Get on/get off" switch.</li> <li>○ Status changes are logged with timestamp and the name of the student.</li> <li>○ If a student is excluded by his/her parents for the current day, that student's status cannot be changed.</li> </ul> </li> <li>– <b>Taking attendance:</b> <ul style="list-style-type: none"> <li>○ Have 2 options: <ul style="list-style-type: none"> <li>▪ Using the list of registered students to take attendance.</li> <li>▪ Using the map to take attendance and also see the live route of the service.</li> </ul> </li> </ul> </li> <li>– <b>Interface:</b></li> </ul>

	<ul style="list-style-type: none"> <li>○ Easy-to-use, quick-access design.</li> </ul>
--	---

## Functional Requirements

Admin Module	<ul style="list-style-type: none"> <li>– Adding A Manager Record: <ul style="list-style-type: none"> <li>○ Entering and saving manager information via the form.</li> </ul> </li> <li>– Delete Manager: <ul style="list-style-type: none"> <li>○ Delete existing manager information.</li> </ul> </li> <li>– Manager List View: <ul style="list-style-type: none"> <li>○ Access to the list of all managers registered to the system.</li> </ul> </li> </ul>
Manager Module	<ul style="list-style-type: none"> <li>– Adding A Service Record: <ul style="list-style-type: none"> <li>○ Entering and saving service information via the form.</li> </ul> </li> <li>– Edit/Delete Service: <ul style="list-style-type: none"> <li>○ Update or delete existing service information.</li> </ul> </li> <li>– Service/Student List View: <ul style="list-style-type: none"> <li>○ Access the list of all services and students registered for the service.</li> </ul> </li> <li>– Reporting: <ul style="list-style-type: none"> <li>○ View time-stamped logs of service movements.</li> </ul> </li> </ul>
Parent Module	<ul style="list-style-type: none"> <li>– Registration and Login: <ul style="list-style-type: none"> <li>○ Secure registration and login processes.</li> </ul> </li> <li>– Adding A Student: <ul style="list-style-type: none"> <li>○ Service selection and entering student information.</li> </ul> </li> <li>– Student Tracking: <ul style="list-style-type: none"> <li>○ View get-in/get-off statuses.</li> </ul> </li> <li>– Student Updates: <ul style="list-style-type: none"> <li>○ Marking a student who will not come, deleting a student or changing the service info.</li> </ul> </li> </ul>
Service Module	<ul style="list-style-type: none"> <li>– Login: <ul style="list-style-type: none"> <li>○ Secure login mechanism.</li> </ul> </li> <li>– Student List View: <ul style="list-style-type: none"> <li>○ View the list of students registered to the service.</li> </ul> </li> <li>– Status Update: <ul style="list-style-type: none"> <li>○ Updating and logging the get-on/land status.</li> </ul> </li> </ul>

	<ul style="list-style-type: none"> <li>– Route Management: <ul style="list-style-type: none"> <li>○ Organizing the student list according to the route.</li> <li>○ Adding useful memos to the students.</li> </ul> </li> </ul>
--	--

## Non-Functional Requirements

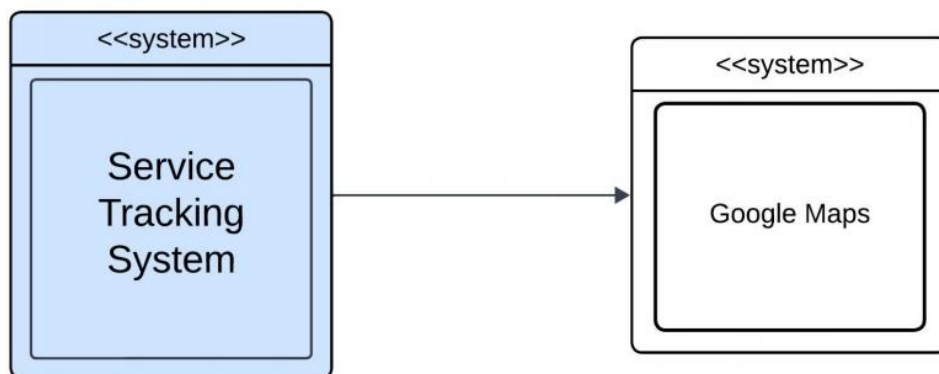
Performance	<ul style="list-style-type: none"> <li>– The system must respond quickly, and user transactions must occur without delay.</li> <li>– It must be resistant to the high number of users.</li> </ul>
Security	<ul style="list-style-type: none"> <li>– User data must be encrypted.</li> <li>– Role-based access controls should be implemented.</li> <li>– There must be data backup and recycling mechanisms.</li> </ul>
Availability	<ul style="list-style-type: none"> <li>– User-friendly interfaces should be designed.</li> <li>– It must be compatible on mobile and web platforms.</li> </ul>
Scalability	<ul style="list-style-type: none"> <li>– Additional schools (as managers) may be added in the future.</li> <li>– The system must be open to expansion.</li> </ul>
Compatibility	<ul style="list-style-type: none"> <li>– It should be compatible with different browsers and devices.</li> <li>– Android support should be considered for mobile applications.</li> </ul>

## Technical Requirements

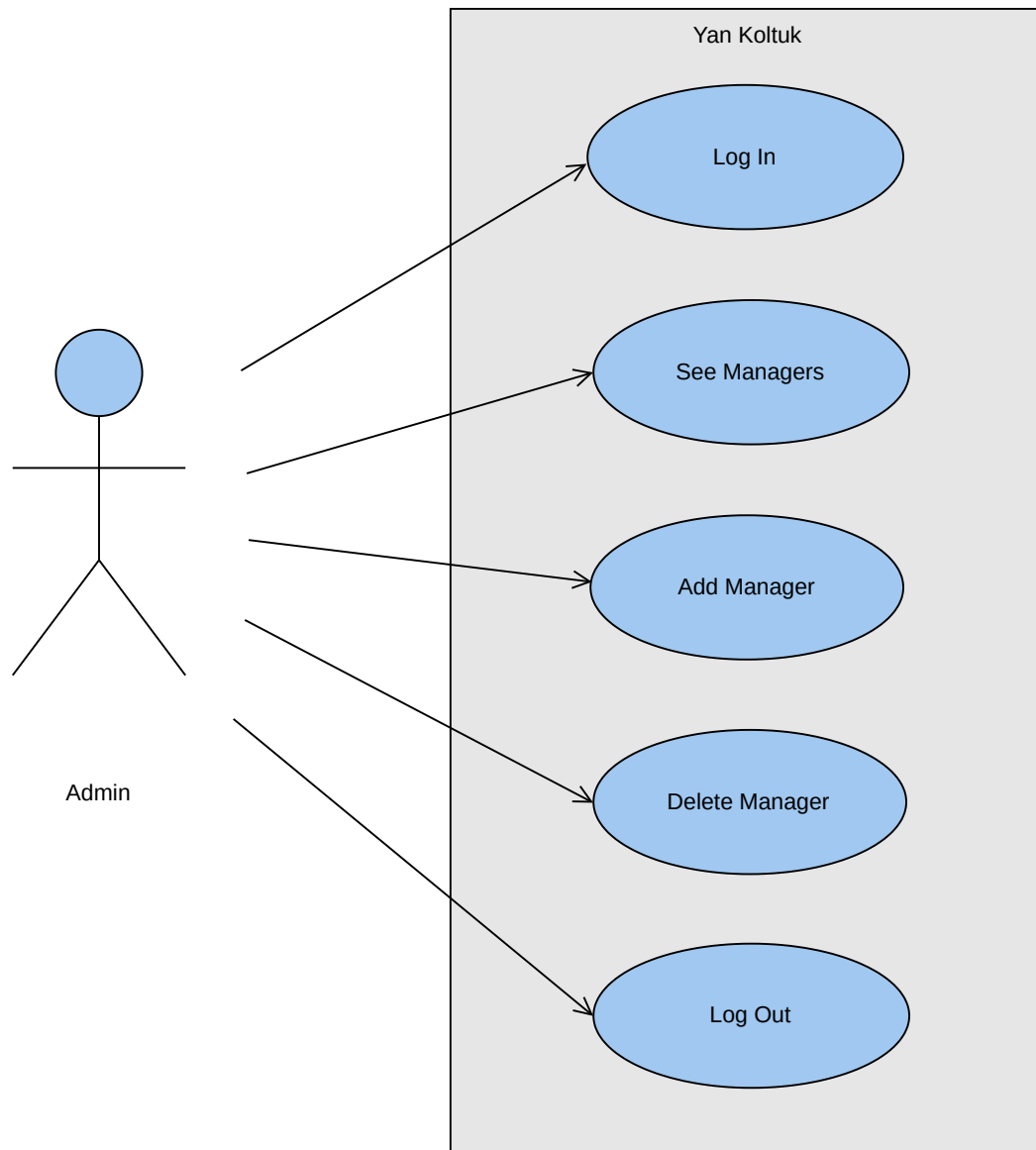
Platforms	<ul style="list-style-type: none"> <li>– Web Application: <ul style="list-style-type: none"> <li>○ Frameworks such as HTML, CSS, JavaScript, React.</li> </ul> </li> <li>– Mobile App: <ul style="list-style-type: none"> <li>○ Flutter</li> </ul> </li> </ul>
Database	<ul style="list-style-type: none"> <li>– Microsoft SQL Server (MSSQL)</li> </ul>

Server and Hosting	<ul style="list-style-type: none"> <li>– Backend: <ul style="list-style-type: none"> <li>○ Asp .NET Core</li> </ul> </li> <li>– Hosting: <ul style="list-style-type: none"> <li>○ Azure (for SQL Server)</li> <li>○ Apache (for deploying Web)</li> </ul> </li> </ul>
Other	<ul style="list-style-type: none"> <li>– Map Integration: <ul style="list-style-type: none"> <li>○ Google Maps API</li> </ul> </li> </ul>

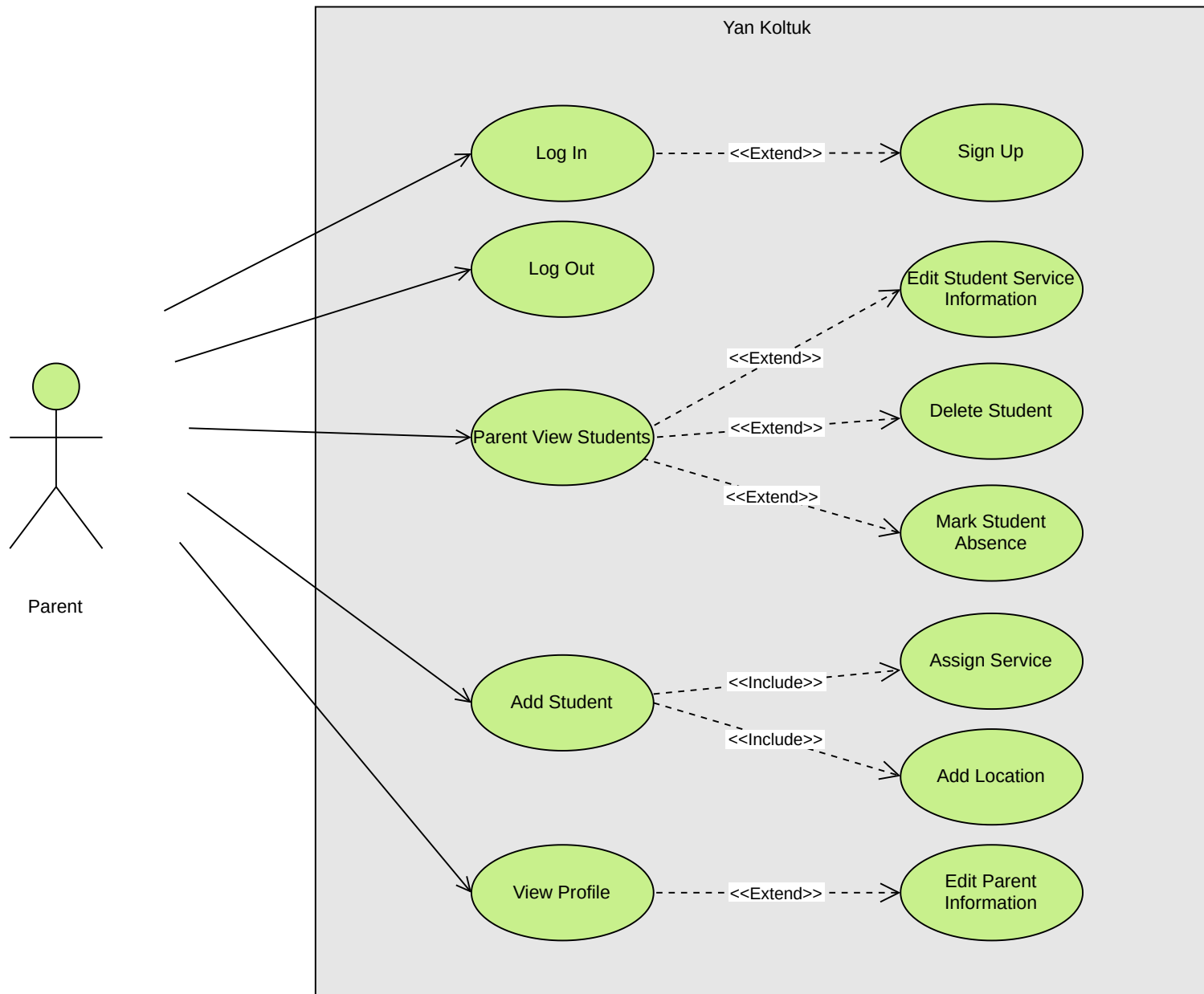
## CONTEXT MODEL

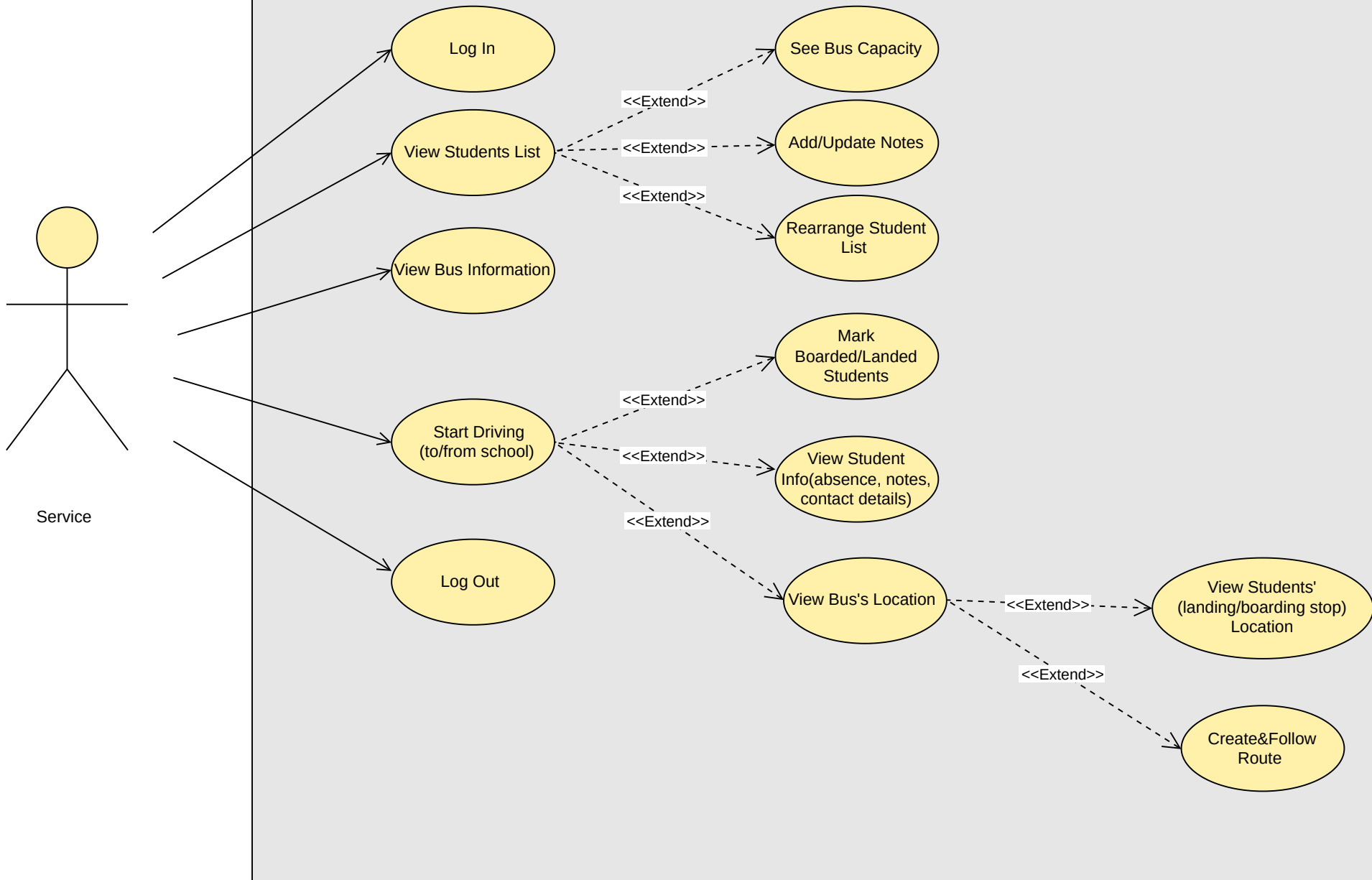












<b>System</b>	Yan Koltuk
<b>Use Case</b>	Log In
<b>Actors</b>	Admin, Parent, Manager, Service
<b>Description</b>	Users can log in to system by filling in required fields
<b>Data</b>	Username and password
<b>Stimulus</b>	The user enters their login credentials and clicks on login button
<b>Response</b>	The user logs in to the system by their role
<b>Comments</b>	The system checks if login credentials are correct or filled.

<b>System</b>	Yan Koltuk
<b>Use Case</b>	Log Out
<b>Actors</b>	Admin, Parent, Manager, Service
<b>Description</b>	Users can log out from the system
<b>Data</b>	None
<b>Stimulus</b>	User clicks on log out button
<b>Response</b>	The user securely exits the system.
<b>Comments</b>	None

<b>System</b>	Yan Koltuk
<b>Use Case</b>	Add Manager
<b>Actors</b>	Admin
<b>Description</b>	Admin can add new managers to the system
<b>Data</b>	Username for new manager
<b>Stimulus</b>	Admin clicks add manager
<b>Response</b>	A password is generated for the new manager and he/she is added to the managers and users table with manager role
<b>Comments</b>	Admin gives the manager his/her password

<b>System</b>	Yan Koltuk
<b>Use Case</b>	See Managers
<b>Actors</b>	Admin
<b>Description</b>	Admin can view managers who use the system
<b>Data</b>	Managers List
<b>Stimulus</b>	Admin clicks "See Managers"
<b>Response</b>	The list of managers is displayed to admin
<b>Comments</b>	none

<b>System</b>	Yan Koltuk
<b>Use Case</b>	Delete Manager
<b>Actors</b>	Admin
<b>Description</b>	Admin can delete a manager from the system

<b>Data</b>	Managers List, Manager Id
<b>Stimulus</b>	Admin selects a manager and clicks “Delete Manager”
<b>Response</b>	The corresponding manager is deleted from Managers List and the screen of the admin updated accordingly
<b>Comments</b>	none

<b>System</b>	Yan Koltuk
<b>Use Case</b>	View Services
<b>Actors</b>	Manager
<b>Description</b>	Manager can view the list of services that are used in his school
<b>Data</b>	Manager Id, Services list ( bus’s plate, driver, driver phone number, stewardess, stewardess’ phone number, capacity, departure time and stop)
<b>Stimulus</b>	Manager clicks view services
<b>Response</b>	The list of services is displayed with their driver and plate information.
<b>Comments</b>	None

<b>System</b>	Yan Koltuk
<b>Use Case</b>	Update Service Information
<b>Actors</b>	Manager
<b>Description</b>	Manager can select one bus from services list and update its details
<b>Data</b>	Manager Id, Services list ( bus’s plate(readonly), driver, driver phone number, stewardess, stewardess’ phone number, capacity, departure time and stop(readonly))
<b>Stimulus</b>	Manager updates one of the textboxes and clicks save button
<b>Response</b>	The corresponding service is updated in database and updated service is displayed to manager.
<b>Comments</b>	This use case is extending “View Services”

<b>System</b>	Yan Koltuk
<b>Use Case</b>	Delete Service
<b>Actors</b>	Manager
<b>Description</b>	Manager can select one bus from services list and delete it
<b>Data</b>	Manager Id, Services list ( bus’s plate(readonly), driver, driver phone number, stewardess, stewardess’ phone number, capacity, departure time and stop(readonly))
<b>Stimulus</b>	Manager selects a bus from services list, see it information in detail and clicks “Delete Service”.

<b>Response</b>	The corresponding service is removed from the manager's services list, and the related column for enrolled students is cleared in the database.
<b>Comments</b>	This use case is extending "View Services"

<b>System</b>	Yan Koltuk
<b>Use Case</b>	View Students
<b>Actors</b>	Manager
<b>Description</b>	Manager can select one bus from services list and see students who are enrolled to that bus
<b>Data</b>	Manager Id, Services table ( bus's plate(readonly), driver, driver phone number, stewardess, stewardess' phone number, capacity, departure time and stop(readonly)), students table( name, surname, id)
<b>Stimulus</b>	Manager clicks a bus from services list
<b>Response</b>	The corresponding service information is displayed in detail, showing students who use that bus
<b>Comments</b>	none

<b>System</b>	Yan Koltuk
<b>Use Case</b>	Add Service
<b>Actors</b>	Manager
<b>Description</b>	Manager can add a new service
<b>Data</b>	Manager Id, Service's information ( bus's plate, driver and stewardess and their phone number, T.C. identity number, bus's capacity, departure time and stop)
<b>Stimulus</b>	Manager clicks "Add Service" then clicks submit button after filling necessary data fields.
<b>Response</b>	If data entered by the manager is in desired format, the new service is created and added to the database.
<b>Comments</b>	none

<b>System</b>	Yan Koltuk
<b>Use Case</b>	View Service Logs
<b>Actors</b>	Manager
<b>Description</b>	The manager can access detailed records of past movements for registered services.
<b>Data</b>	Service Id, Service Logs and Students table
<b>Stimulus</b>	Manager clicks on "View Service Logs"
<b>Response</b>	The manager views service reports from the last two weeks, including students' details and their boarding/landing times.
<b>Comments</b>	This case extended by "Generate Reports"

<b>System</b>	Yan Koltuk
<b>Use Case</b>	Generate Reports
<b>Actors</b>	Manager
<b>Description</b>	The manager can download the service logs as a document.
<b>Data</b>	Service Id, Service Logs and Students table
<b>Stimulus</b>	Manager clicks on “Download Report”
<b>Response</b>	A pdf file is created and downloaded.
<b>Comments</b>	none

<b>System</b>	Yan Koltuk
<b>Use Case</b>	Sign Up
<b>Actors</b>	Parent
<b>Description</b>	Parent can sign up to the system to register their student to a service etc.
<b>Data</b>	Name, Identity Number, Phone number, Password, Address
<b>Stimulus</b>	Parent enters the required credentials
<b>Response</b>	Parent is successfully registered to the system
<b>Comments</b>	Parent is registered if they entered valid data.

<b>System</b>	Yan Koltuk
<b>Use Case</b>	Parent View Students
<b>Actors</b>	Parent
<b>Description</b>	Parents can view the list of registered students and their boarding/landing status, edit their information in this screen.
<b>Data</b>	Student list
<b>Stimulus</b>	Parent logs in to the system
<b>Response</b>	Parents can view their student list
<b>Comments</b>	If parent want to make changes of students’ status -like attendance-, they will do that also in that screen.

<b>System</b>	Yan Koltuk
<b>Use Case</b>	Edit Student Information
<b>Actors</b>	Parent
<b>Description</b>	Parent can change the service of the student by selecting a plate
<b>Data</b>	New plate is added to student service table
<b>Stimulus</b>	Parent select a plate for the student and clicks on save button
<b>Response</b>	System saves the changed data and assigns it for the student
<b>Comments</b>	This use case is extending “Parent View Students”



<b>System</b>	Yan Koltuk
<b>Use Case</b>	Delete Student
<b>Actors</b>	Parent
<b>Description</b>	When ac student will not take the service anymore parent can delete the corresponding student.
<b>Data</b>	Selected student is deleted from student service table
<b>Stimulus</b>	Parent selects the student that they want to delete it from the system and chooses the delete student option
<b>Response</b>	System successfully deletes the student
<b>Comments</b>	None

<b>System</b>	Yan Koltuk
<b>Use Case</b>	Mark Student Absence
<b>Actors</b>	Parent
<b>Description</b>	If a student will be absent, their parent can mark them as such before the service departs, avoiding unnecessary stops and delays.
<b>Data</b>	The date interval of absence of student for the service
<b>Stimulus</b>	Parent chooses the student for marking attendance, clicks on mark attendance button and selects an interval or just one day-off from the calendar for the student.
<b>Response</b>	The date interval of absence of student for the service is inserted to student service table. System marks the student as not coming on that interval of dates and driver can see absence with “View Student List” option, this preserves driver from unnecessary waitings on the location for that student get on the service.
<b>Comments</b>	Parent can see the selected dates in calendar and update the remaining excluded dates if need be.

<b>System</b>	Yan Koltuk
<b>Use Case</b>	Add Student
<b>Actors</b>	Parent
<b>Description</b>	Parent can add new student to the system to track him/her status
<b>Data</b>	Student’s name, surname, school id, T.C. identity number, assigned service.
<b>Stimulus</b>	Parent clicks save button
<b>Response</b>	The student is successfully added to Students table.
<b>Comments</b>	none

<b>System</b>	Yan Koltuk
<b>Use Case</b>	Assign Service
<b>Actors</b>	Parent
<b>Description</b>	When parents add a student to the system, he/she has to choose a service within the available list.
<b>Data</b>	Plate number of the service, student service table
<b>Stimulus</b>	Parent chooses the service plate on the combo box
<b>Response</b>	The student is successfully registered to the selected service
<b>Comments</b>	Student is registered if only there is enough capacity in the service bus.

<b>System</b>	Yan Koltuk
<b>Use Case</b>	Add Location
<b>Actors</b>	Parent
<b>Description</b>	When parents add a student to the system, he/she has to choose a location on maps to save the location of the boarding/landing stop of the student.
<b>Data</b>	Latitude and longitude information of students is saved to database
<b>Stimulus</b>	Parent chooses “Add Location”
<b>Response</b>	The student is successfully registered to the selected service with location information
<b>Comments</b>	none

<b>System</b>	Yan Koltuk
<b>Use Case</b>	View Profile
<b>Actors</b>	Parent
<b>Description</b>	Parent can view his/her own information.
<b>Data</b>	Parent’s name, surname, address from the parents table.
<b>Stimulus</b>	Parent clicks on view profile button
<b>Response</b>	Parent can view their information on the system
<b>Comments</b>	Parent can edit their information also in this screen

<b>System</b>	Yan Koltuk
<b>Use Case</b>	Edit Parent Information
<b>Actors</b>	Parent
<b>Description</b>	Parent may update his/her information
<b>Data</b>	-Phone number, Address -some of them that will be changed-
<b>Stimulus</b>	Parent clicks on change button and enters the data that will be changed to
<b>Response</b>	Parent can successfully change their information, parents table is updated

<b>Comments</b>	Parent can update the information on system if given data is valid, like data does not exceed the character limit, or phone number doesn't have alphabetic characters in it.
-----------------	--

System	Yan Koltuk
Use Case	View Students List
Actors	Service
Description	Students who are enrolled to a bus can be viewed as a list in mobile app.
Data	Students table, student service table, parents table, service table. Student services are accessed with service id, then student ids are accessed, with student ids parent and student information is accessed.
Stimulus	Service clicks "Student List" on mobile app
Response	Students' names, parent's names and contact details and notes which were added by service are displayed on mobile screen. If a student is selected, Service can see other details such as address and students schoolId on read-only mode.
Comments	This use case extended by tree other use cases: See Bus Capacity, Add/Update Notes, Rearrange Student List

System	Yan Koltuk
Use Case	See Bus Capacity
Actors	Service
Description	Service can view the number of students enrolled / bus's capacity
Data	Service table, Student Service table. Corresponding service is found from service id, capacity is fetched. The number of student services whose service id is same with the current service's id is counted.
Stimulus	Service clicks "Student List" on mobile app
Response	The occupancy rate is displayed on the up-right-corner.
Comments	This use case is extends "View Students List"

System	Yan Koltuk
Use Case	Rearrange Students List
Actors	Service

Description	Service can change the order of students, this order will be saved. When service starts driving, names and details of students will be displayed in this order.
Data	Student Service table
Stimulus	Service press and holds a student component, then service becomes able to place it to the desired position
Response	Student Services are reorganized both in database and mobile app.
Comments	This use case is extends "View Students List"

System	Yan Koltuk
Use Case	Add/Update Notes
Actors	Service
Description	Service can add a note to a student or can update an existing notes. When service starts driving these notes are displayed in order to have quick reminders about students' boarding/landing stops <i>e.g. in front of Flower Bakery</i>
Data	Student Service table
Stimulus	Service clicks on edit button(pen image)
Response	Note is inserted/updated to Student Service table
Comments	This use case is extends "View Students List".

System	Yan Koltuk
Use Case	Start Driving (to/from school)
Actors	Service
Description	In order to start the ride Service should choose one of these options: Departure to School, Return from School. Then Service will be able to mark students boarding status.

Data	The trip type is preserved according to the Service's choice
Stimulus	Service clicks Departure to School or Return from School
Response	Trip Type is updated accordingly, a new screen that extends two features is displayed.
Comments	This use case is extended by "Mark Boarded/Landed Students" and "View Student Info(absence, notes, contact details)".

System	Yan Koltuk
Use Case	View Student Info(absence, notes, contact details)
Actors	Service
Description	When the service begins driving, a pre-ordered list of students is displayed, sorted by the sequence of their stops. Students marked absent by their parents are highlighted in a distinct color, allowing the service to identify and skip unnecessary stops. The service can also view parents' contact details directly, enabling seamless communication during unexpected situations without switching screens. Additionally, notes about each student help the service locate and recognize them more easily. This use case simplifies the driving process, ensuring efficiency and smooth navigation.
Data	Student Service, Students, Parents, Service tables. Student services which has current services id is fetched, then with their student ids parent and student details are accessed.
Stimulus	Service clicks Departure to School or Return from School
Response	The students list is displayed
Comments	This use case extends "Start Driving (to/from school)"

System	Yan Koltuk
Use Case	Mark Boarded/Landed Students
Actors	Service
Description	After boarding or landing students service turns on a student's switch to mark him as landed or boarded.

Data	Student Service, Service tables. Student Services are accessed via current Service's id, then students are accessed with names and GetOff_GetOn coloumn is updated accordingly
Stimulus	Service turns on switches
Response	Students' boarding/landing status are updated parents can see it via mobil app.
Comments	This use case extends "Start Driving (to/from school)"

System	Yan Koltuk
Use Case	View Bus Information
Actors	Service
Description	Driver's and Stewardess's name and contact details, bus's plate and departure information is displayed on the mainpage of the mobile app
Data	Information is fetched from Service table
Stimulus	Service logs in to the mobile app
Response	Details about bus is displayed
Comments	none

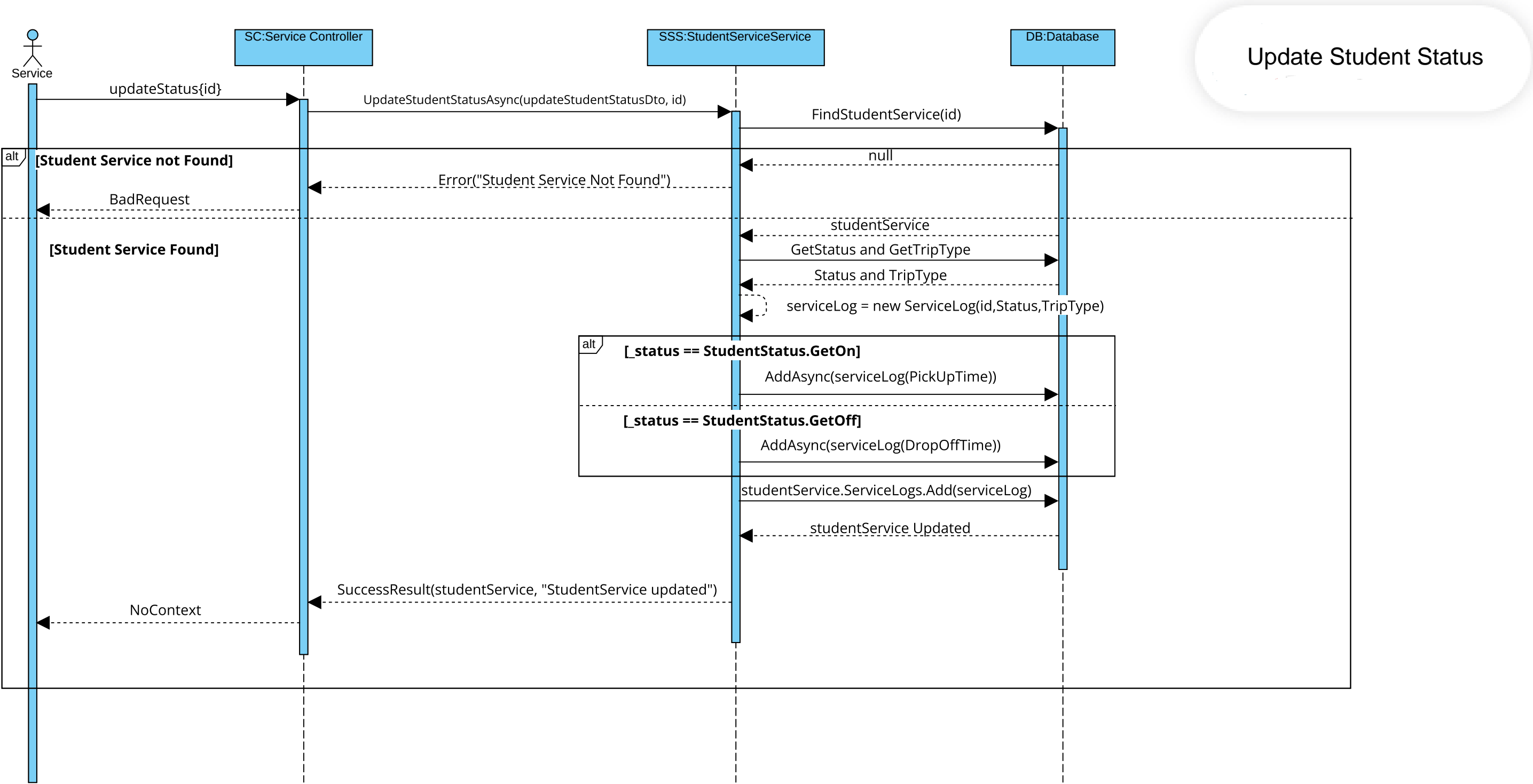
System	Yan Koltuk
Use Case	View Bus's Location
Actors	Service
Description	Service can see it's own location on Google Maps
Data	Location( latitude and longtitude information of service)
Stimulus	Service clicks "View Location"
Response	Bus's Live Location is displayed on Maps.

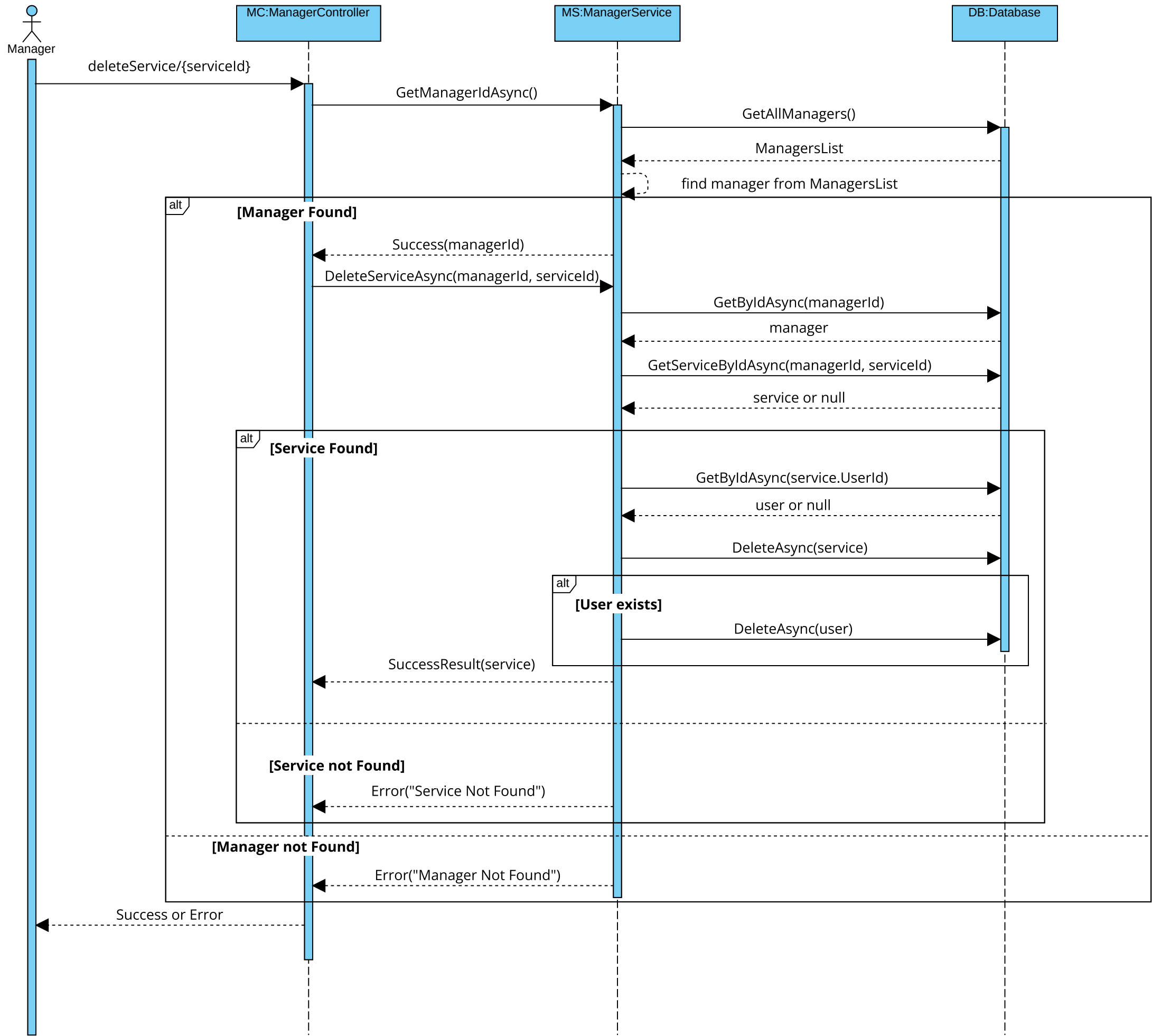
Comments	This use case is extended by “View Students' (landing/boarding stop) Location” and “Create&Follow Route”
----------	--

System	Yan Koltuk
Use Case	View Students' (landing/boarding stop) Locations
Actors	Service
Description	The service can view predefined stop locations marked on Google Maps alongside the real-time bus location. This stops were defined by parents when they added students.
Data	Stop location information of students( latitude and longitude values) from Students table
Stimulus	Service clicks “View Students’ Locations”
Response	Predefined stops are marked on Google Maps
Comments	This use case extends “View Bus’s Location”

System	Yan Koltuk
Use Case	Create&Follow Route
Actors	Service
Description	The service can create a route with predefined stops and follow it seamlessly with Google Maps.
Data	Index of students
Stimulus	Service clicks “Create Route”
Response	Route is displayed on map with the live location of the bus and stops’ locations
Comments	This use case extends “View Bus’s Location”. If the bus is returning from school the students list is reversed and route created accordingly.

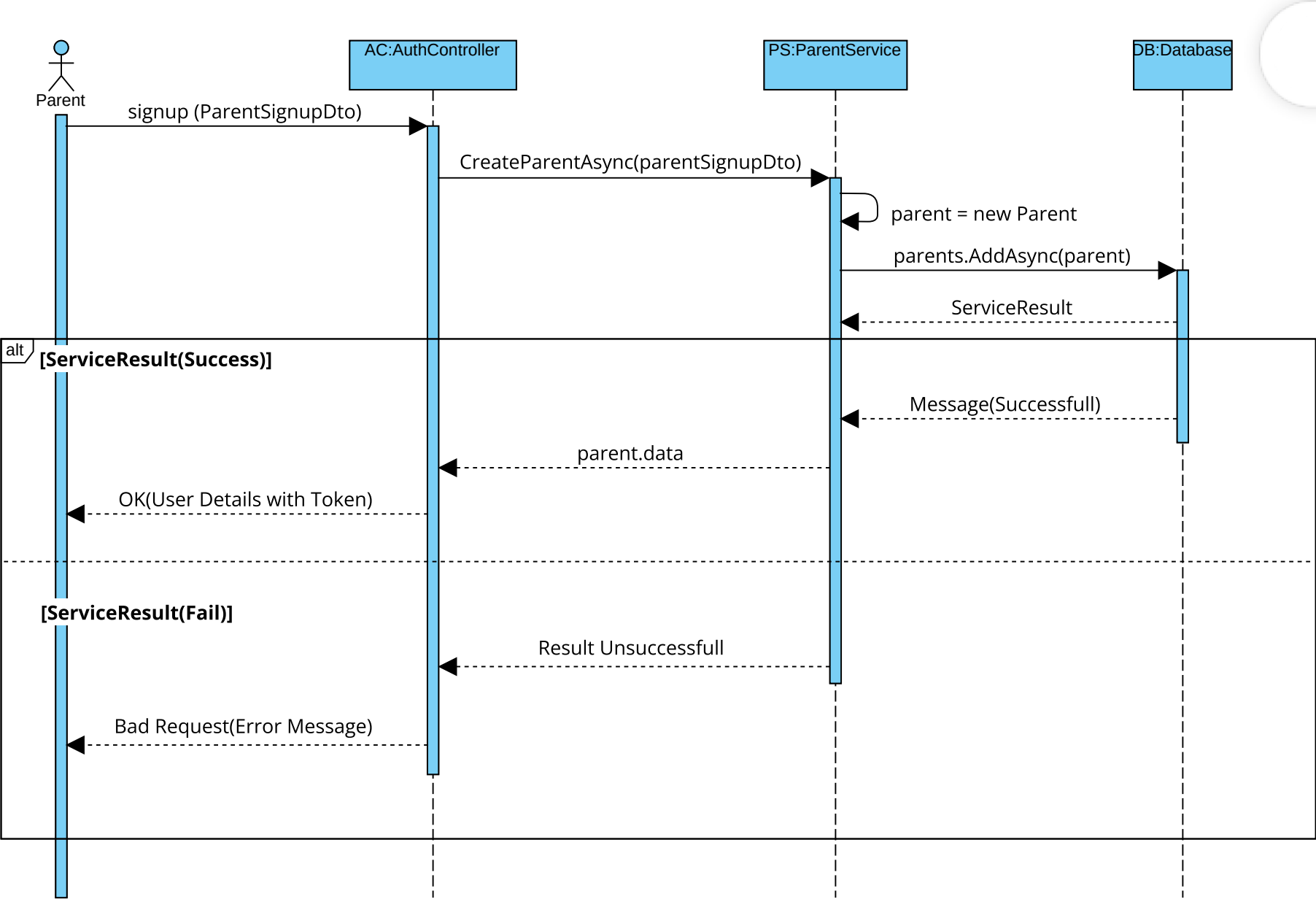


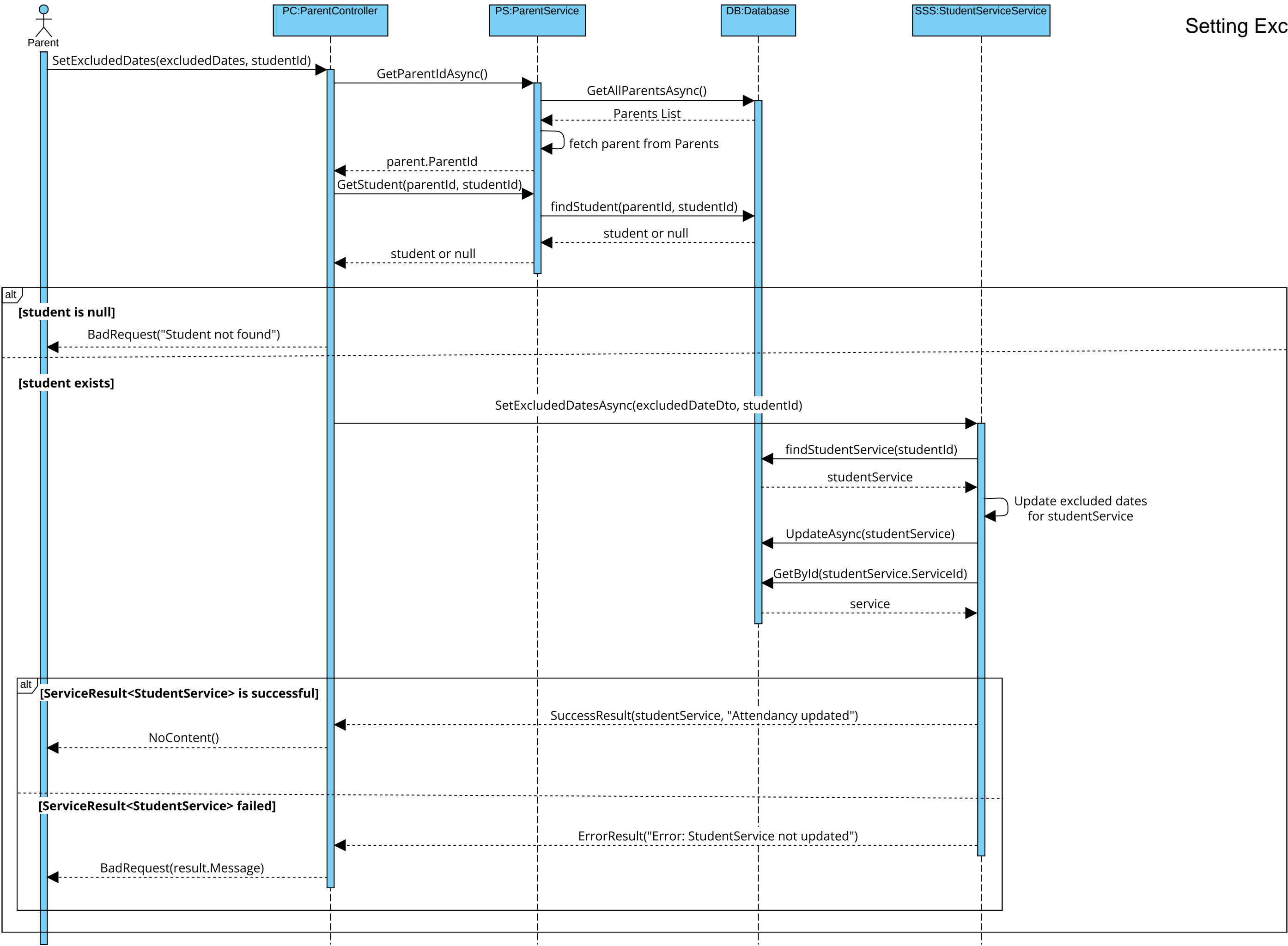


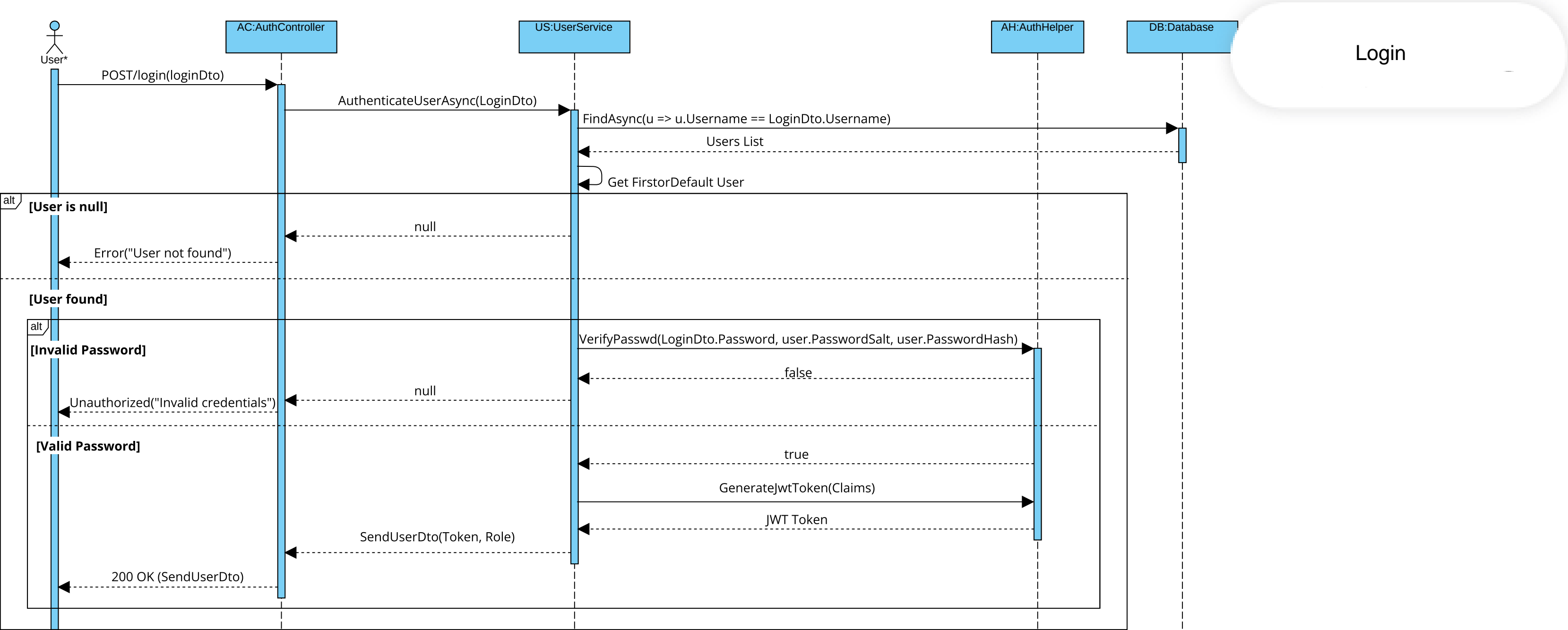


# Delete Service

Sign Up







\*: All actors of system are also saved as users with different roles in database. Log in process is same for all users.