Insurance Purchase System (Personal Auto)

Phase 2 - System and database design

Debashis Jena

University of Maryland University College

Table of Contents

[A. User Interface (Self-Service Personal Auto Insurance Purchase System) 2](#_Toc25999813)

[B. Database Design 14](#_Toc25999814)

[C. System architecture 23](#_Toc25999815)

[References 26](#_Toc25999816)

# User Interface (Self-Service Personal Auto Insurance Purchase System)

As discussed in the Phase 1 of the project, the proposed application is going to be customer facing. The selfservice insurance application order form will have a landing page, whose link will be available in the company’s website. The landing page link might be sent to the user as part of a marketing email or a direct mail.

In the phase 1, the context diagram represents the system’s internal processes, datasources, and how it is going to communicate with the external systems or databases. Figure A.1 is a high level screenflow architecture of the self service system. The quoting application will have the first external API integration with MVR, where the data about the customer’s driving history will be retrieved for underwriter’s review. Based on the underwriter’s approval decision, customer will be presented with the next pages. Upon approval, payment method information will be requested to enter. Payment methods are processed in an existing billing system. The customer will be fulfilled with a confirmation email with links to download the documents and open a new online account.

A screenshot of a cell phone

Description automatically generated

Figure A.1 High-level design of the self-service system.

1. Screen-By-Screen flow

The personal auto insurance line of business involves many terms and jargons which customers may not be very familiar with. Therefore, it’s critical for the screen navigation to be self explanatory. Therefore, the application will be screen to screen navigation using “Next” and “Back” buttons.

Figure A.2. gives the holistic view of the screen-by-screen navigation. The interconnecting arrows represent the navigation from one page to the other.

A close up of a logo

Description automatically generated

Figure A.2 Screen flow

1. Individual Screen Requirements –

The user input in the web page must satisfy the below checklists.

* Completeness – Verify if the required fields are complete. If the Next button is clicked and any of the required fields is/are incomplete, then an error message saying “<Field label> is required.”
* Format – Validate the entered value against the format allowed in the field. An alphabet only field must not allow numbers or special characters. Likewise, a number only field must not allow any other type of data.
* Database validations – Datbase validations are required when a duplicate policy is being created for the same user email and SSN combination. Also, the database validation is required while the online account is being created to check if the user name currently exists.

The page descriptions are as follows.

* 1. Named Insured

This is the first page where the basic information of the first named insured is collected. Optionally, the customer may want to add one more named insured, who may be a spouse or a friend or relative of the customer. A field with an asterisk represents a required field

* + - 1. First Name – input field, alphanumeric
      2. Middle Name – input field, alphanumeric
      3. Last Name – input field, alphanumeric
      4. Gender – picklist with gender options
      5. Marital Status – picklist with marital status options
      6. Address 1 (Street address) – input field, alphanumeric
      7. Address 2 (Apt #, Ste #, etc) – input field, alphanumeric
      8. City – input field, alphanumeric
      9. State – picklist with abbreviated codes of all the states
      10. Zip Code – input field, 5 digit numeric
      11. E-mail – input field, alphanumeric, email text validation
      12. Birth Date – date picker
      13. Social Security Number – input field, 9 digit numeric

|  |  |
| --- | --- |
| **Named Insured Information** | |
| **First Name** | **\*** |
| **Middle Name** |  |
| **Last Name** | **\*** |
| **Gender** | \* |
| **Marital Status** | \* |
| **Address 1 (Street address)** | **\*** |
| **Address 2 (Apt #, Ste #, etc)** |  |
| **City** | **\*** |
| **State** | **\*** |
| **Zip Code** | \* |
| **E-mail** | \* |
| **Birth Date** | (e.g. mm-dd-yyyy) \* |
| **Social Security Number** | **\*** |
| **Next >>**  **<<Back**  **Cancel** | |

* 1. Driver Information

As the Next button is clicked in the Named Insured Information page, the customer lands in the Driver Information page. As default drivers, the first and second named insured will be listed. This page allows to have more drivers apart from the default ones.

* + - 1. First Name – input field, alphanumeric
      2. Middle Name – input field, alphanumeric
      3. Last Name – input field, alphanumeric
      4. Gender – picklist with gender options
      5. Marital Status – picklist with marital status options
      6. Birth Date – date picker
      7. State Licensed – picklist with abbreviated codes of all the states
      8. Driver’s License Number – input field, alphanumeric
      9. Has the driver completed an approved Accident Prevention course within the last 3 years? – radio buttons, yes / no
      10. Has the driver completed a Driver (training) Education course? – radio buttons, yes / no

|  |  |
| --- | --- |
| **Driver Information** | |
| *Enter the name as it appears on the driver’s license.* | |
| **First Name** | **\*** |
| **Middle Name** |  |
| **Last Name** | **\*** |
| **Gender** | \* |
| **Marital Status** | \* |
| **Birth Date** | (e.g. mm-dd-yyyy) \* |
| **State Licensed** | **\*** |
| **Driver’s License Number** | **\*** |
| **Has the driver completed an approved Accident Prevention course within the last 3 years?** | \* |
| **Has the driver completed a Driver (training) Education course?** | \* |
| **Add Driver**  **Next >>**  **<<Back**  **Cancel** | |

* 1. Vehicle Information –

Multiple vehicles can be added in this page.

* + - 1. Year – picklist with years as options
      2. Make – picklist with car brand names, Honda, Toyota, etc
      3. Model – as the Make is selected, this picklist gets updated with relevant options for the corresponding Make. For example, if Honda is selected, the options available here would be Civic, Accord, CRV, etc.
      4. VIN – input field, alphanumeric
      5. Vehicle Registered State – picklist with abbreviated codes of all the states
      6. Anti-Lock Brakes – radio buttons, yes / no
      7. Anti-Theft Device – radio buttons, yes / no
      8. Is this car owned? – radio buttons, yes / no
      9. Name of the Financing Company – If the previous field is selected as Yes, then this field displays and is required.

|  |  |
| --- | --- |
| **Vehicle Information** | |
| **Year** | **\*** |
| **Make** | **\*** |
| **Model** | **\*** |
| **VIN** | \* |
| **Vehicle Registered State** | \* |
| **Anti-Lock Brakes** | **\*** |
| **Anti-Theft Device** | **\*** |
| **Is this car owned?** | **\*** |
| **Name of the Financing Company** | \* |
| **Add Vehicle**  **Next >>**  **<<Back**  **Cancel** | |

* 1. Liability Coverage Information

This is a page where policy level coverage is selected. These coverage limits are applicable for all drivers and named insureds listed on the policy. As the customer selects Next button in the Vehicle Information page, the below page is displayed.

* + - 1. Bodily Injury Liability – Picklist with options like $100K/$300K, $300K/$300K.
      2. Property Damage Liability – Picklist with options like $50K, $100K, $150K, $200K.
      3. Uninsured Motorist Bodily Injury – This field is prepopulated with the selection from the “Bodily Injury Liability” field. However it can be overridden with a higher value.
      4. Uninsured Motorist Property Damage – This field is prepopulated with the selection from the “Property Damage Liability” field. However it can be overridden with a higher value.

|  |  |
| --- | --- |
| **Liability Coverage Information** | |
| **Bodily Injury Liability** | **\*** |
| **Property Damage Liability** | **\*** |
| **Uninsured Motorist Bodily Injury** | **\*** |
| **Uninsured Motorist Property Damage** | \* |
| **Next >>**  **<<Back**  **Cancel** | |

* 1. Physical Damage Coverage Information

Physical Damage is a vehicle specific coverage. That’s why all the vehicles are listed in this page for the coverage limits to be selected.

* + - 1. Select Vehicle – picklist with list of vehicle description like, 2012-Honda-Civic, 2018-BMW-X5
      2. Comprehensive Deductible – picklist with options like $250, $500
      3. Collision Deductible – picklist with options like $250, $500
      4. Rental Reimbursement Coverage – picklist options like $20/$600, $30/$900

Unlike previous pages, this page will not have a Next button, rather it will have a Calculate Premium button. That gives a hint to the user that this is the last screen to enter all the policy related information.

|  |  |
| --- | --- |
| **Physical Damage or Vehicle Coverage Information** | |
| **Select Vehicle** | **\*** |
| **Comprehensive Deductible** | **\*** |
| **Collision Deductible** | **\*** |
| **Rental Reimbursement Coverage** | \* |
| **Calculate Premium >>**  **Cancel**  **<<Back** | |

* 1. Quote / Premium

As the Calculate Premium button in the previous page is clicked, the customer lands in this page and a spinner displays indicating the system doing the below activities in the background.

1. Getting base rates for the coverage selected based on the State (Insurance rates differ from state to state)
2. Calculating the total premium
3. Deducting the applicable discounts, such as Safe Driver discount, Good Student discount
4. Making the API call to the MVR (Motor Vehicle Record) for the driver’s history
5. Also, underwriter (UW) approval rules evaluation for the preliminary review

Once, all of the above actions are complete the system displays the premium if the preliminary UW approval is received. If the customer information is found to be ineligible in the UW review, a “We’re Sorry” page is displayed and that ends the customers session.

|  |  |  |  |
| --- | --- | --- | --- |
| **Total Premium** | | | |
|  | Coverage | Vehicle 1 | Vehicle 2 |
| **Bodily Injury Liability** | $100,000/$300,000 | $60.00 | $40.00 |
| **Property Damage Liability** | $50,000 | $100.00 | $65.00 |
| **Personal Injury Protection** | Option A | $3.00 | $3.00 |
| **Uninsured Motorists Bodily Injury** | $100,000/$300,000 | $15.00 | $10.00 |
| **Uninsured Motorists Property Damage** | $50,000 | $5.00 | $3.00 |
| **Vehicle Level Coverage** | | | |
| **Comprehensive** | $500 Ded | $25.00 | $16.00 |
| **Collision** | $500 Ded | $70.00 | $45.00 |
| **Emergency Road Service** | Full | $2.00 | $3.00 |
| **Mechanical Breakdown** | $250 Ded | $20.00 | $15.00 |
| **Six Month Premium Per Vehicle** | | **$300.00** | **$200.00** |
| **Cancel**  **<<Edit Coverage**  **Purchase Now >>** | | | |

* 1. Payment

Upon preliminary UW approval, system displays the calculated premium with two new buttons; Edit Coverage and Purchase Now. If the customer needs to change the coverage, then Edit Coverage button can be clicked to go back to the previous pages and make the changes. If customer is happy with the premium, Purchase Now button can be clicked to go to the continue with the new business and convert the quote into an actual policy.

In this page, system displays the payment related field. Currently, the company accepts electronic payments only. So the below fields are displayed for credit or debit card information.

1. Credit Card Type – picklist with options like Visa, AmEx, MasterCard
2. Credit Card Number – 15 or 16 digit credit card number
3. Expiration Month – list of months
4. Expiration Year – list of next 15 years, starting from current year
5. Credit Card CVV – input field, accepts 3 or 4 digits.

|  |  |
| --- | --- |
| **Payment** | |
| **Payment Amount** | **$500.00** ($300.00 + $200.00) |
| **Credit Card Type** | **\* (Visa, Amex, etc.)** |
| **Credit Card Number** | \* |
| **Expiration Month** | \* |
| **Expiration Year** | \* |
| **Credit Card CVV** | \* |
| **Submit Payment>>**  **<<Back**  **Cancel** | |

* 1. Create Account

As the payment is processed in the previous page, the new page displays where the system requires the customer to create an account where the policy related information can be viewed or edited. Also, the insurance id card and the declaration page can be downloaded or printed from the online account.

1. Email – this field is already prepopulated with the user email from the first page
2. Password – input field
3. Re-enter password – input field, must match with the previous field.

|  |  |
| --- | --- |
| **Create Online Account** | |
| **Email** | \* |
| **Enter Password** | \* |
| **Re-Enter Password** | \* |
| **Reset**  **Create Account >>** | |

As the Create Account button in the last page is clicked, an authentication record is created for the customer in the database and the policy is connected with this online account as a related record in the background. Customer lands in the home page of the online account.

This completes the selfservice application form navigation. Customer will not be able to go back to any of the previous pages any longer. For any policy related changes, the online account should be accessed to achieve this.

# Database Design

1. Normalized Database Design

A typical personal auto insurance policy captures the below datapoints. The main factors to calculate the premium are, the information about the driver, the vehicles and the coverages selected. On a high level the below sample data can be a set of data required for a personal auto policy.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| cust  num | name | driver | vehicle | Liability  covg | Physical  damage | policy\_id | invoice\_id | payment\_id | payment\_method |
| 1 | John Doe | John Doe | 2014-Honda-civic | 100,000 | 20,000 | p1 | inv001 | pmt001 | pmid001 |
|  |  | Jane Doe | 2017-Nissan-Murano |  | 25,000 |  | inv002 | pmt001 | pmid001 |
|  |  | John Doe | 2015-BMW-X1 |  | 25,000 |  | inv003 | pmt001 | pmid001 |
| 2 | Ram Kumar | Ram Kumar | 2019-Toyota-Camry | 300,000 | 25,000 | p2 | inv004 | pmt002 | pmid002 |
|  |  | Laxman Kumar | 2018-Audi-Q7 |  | 50,000 |  | inv004 | pmt003 | pmid003 |

(Customer Id, Customer Name, [Driver, Vehicle, Liability Coverage, Physical Damage Coverage, Policy Id, Invoice Id, Payment Id, Payment Method Id]).

The above data is unnormalized. To normalize the three factor normalization steps can be followed.

1. **1NF - Remove all repeating groups and identify the primary key** – Reduce the first level of redundancy. Customer table can be moved out and be a separate table, then the customer Id can be the primary key.

|  |  |
| --- | --- |
| cust\_num | name |
| 1 | John Doe |
| 2 | Ram Kumar |

Customer (cust\_num, Name)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Cust\_id | driver | vehicle | liability\_covg | physical\_damage | policy\_id | invoice\_id | payment\_id | payment\_method |
| 1 | John Doe | 2014-Honda-civic | 100,000 | 20,000 | p1 | inv001 | pmt001 | pmid001 |
| 1 | Jane Doe | 2017-Nissan-Murano | 100,000 | 25,000 | p1 | inv002 | pmt001 | pmid001 |
| 1 | John Doe | 2015-BMW-X1 | 100,000 | 25,000 | p1 | inv003 | pmt001 | pmid001 |
| 2 | Ram Kumar | 2019-Toyota-Camry | 300,000 | 25,000 | p2 | inv004 | pmt002 | pmid002 |
| 2 | Laxman Kumar | 2018-Audi-Q7 | 300,000 | 50,000 | p2 | inv004 | pmt003 | pmid003 |

Policy (Cust\_Id, Driver, Vehicle, Liability Coverage, Physical Damage Coverage, Policy Id, Invoice Id, Payment Id, Payment Method Id)

1. **2NF - All non-key attributes to be fully dependent on the primary key** – In the policy table the driver is repeating, since a driver can be assigned to multiple vehicles. Similarly, since the liability coverage is customer specific and is repeating in the first step, it can be moved to a separate table.

|  |  |
| --- | --- |
| cust\_num | name |
| 1 | John Doe |
| 2 | Ram Kumar |

|  |  |  |
| --- | --- | --- |
| driver\_id | Cust\_id | Driver\_name |
| drv1 | 1 | John Doe |
| drv2 | 1 | Jane Doe |
| drv3 | 2 | Ram Kumar |
| drv4 | 2 | Laxman Kumar |

|  |  |
| --- | --- |
| customer\_id | Liability\_coverage |
| 1 | 100,000 |
| 2 | 300,000 |

Customer (cust\_num, Name) Driver(driver\_id, cust\_id, driver\_name)

Liability Coverage(cust\_id, Liability coverage amount)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Cust\_id | vehicle | physical\_damage | policy\_id | invoice\_id | payment\_id | Payment  method |
| 1 | 2014-Honda-civic | 20,000 | p1 | inv001 | pmt001 | pmid001 |
| 1 | 2017-Nissan-Murano | 25,000 | p1 | inv002 | pmt001 | pmid001 |
| 1 | 2015-BMW-X1 | 25,000 | p1 | inv003 | pmt001 | pmid001 |
| 2 | 2019-Toyota-Camry | 25,000 | p2 | inv004 | pmt002 | pmid002 |
| 2 | 2018-Audi-Q7 | 50,000 | p2 | inv004 | pmt003 | pmid003 |

Policy (Cust\_Id, Vehicle, Physical Damage Coverage, Policy Id, Invoice Id, Payment Id, Payment Method Id)

1. **3NF - Remove any transitive dependencies** – After two steps it can be noticed that there are still some redundancies in the invoice and payment columns which look to be many-to-many relationship. In addition, more datapoints for vehicle, coverage and payme t method would be required to be captured. Here is a simplified 3NF table design for this application.

|  |  |  |
| --- | --- | --- |
| driver\_id | Cust\_id | Driver\_name |
| drv1 | 1 | John Doe |
| drv2 | 1 | Jane Doe |
| drv3 | 2 | Ram Kumar |
| drv4 | 2 | Laxman Kumar |

|  |  |
| --- | --- |
| cust\_num | name |
| 1 | John Doe |
| 2 | Ram Kumar |

Customer (cust\_num, Name) Driver(driver\_id, cust\_id, driver\_name)

|  |  |
| --- | --- |
| customer\_id | Liability\_coverage |
| 1 | 100,000 |
| 2 | 300,000 |

LiabilityCoverage(cust\_id, Liability coverage amount)

|  |  |  |  |
| --- | --- | --- | --- |
| Vehicle\_Id | cust\_num | vehicle | mileage |
| v1 | 1 | 2014-Honda-civic | 90,000 |
| v2 | 1 | 2017-Nissan-Murano | 28,000 |
| v3 | 1 | 2015-BMW-X1 | 40,000 |
| v4 | 2 | 2019-Toyota-Camry | 6,000 |
| v5 | 2 | 2018-Audi-Q7 | 14,000 |

Vehicle(Vehicle\_Id, cust\_Id, vehicle\_information, mileage)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| vehicle\_id | Coverage\_id | physical\_damage | collision\_ded | comprehensive\_ded |
| v1 | pd1 | 20,000 | 250 | 500 |
| v2 | pd2 | 25,000 | 250 | 1,000 |
| v3 | pd3 | 25,000 | 250 | 500 |
| v4 | pd4 | 25,000 | 500 | 500 |
| v5 | pd5 | 50,000 | 250 | 250 |

PhysicalDamage (vehicle\_Id, coverage\_id, physical\_damage\_amount, collision\_ded, comp\_ded)

|  |  |
| --- | --- |
| Cust\_id | policy\_id |
| 1 | p1 |
| 1 | p1 |
| 1 | p1 |
| 2 | p2 |
| 2 | p2 |

Policy (Cust\_Id, Policy Id)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| cust\_num | payment\_method | card\_type | card\_num | exp\_date | status |
| 1 | pmid001 | visa | \*\*\*\*\*\*1111 | 11/21 | Active |
| 2 | pmid002 | visa | \*\*\*\*\*\*2121 | 01/22 | Active |
| 2 | pmid003 | Amex | \*\*\*\*\*\*213 | 10/23 | Active |

Policy (Cust\_Id, Payment Method Id, Credit Card (CC) Type, CC Number, CC Exp, CC Status)

|  |  |  |
| --- | --- | --- |
| policy\_id | invoice\_id | invoice\_amount |
| p1 | inv001 | 250 |
| p1 | inv002 | 450 |
| p1 | inv003 | 300 |
| p2 | inv004 | 1200 |

Invoice (policy\_Id, Invoice Id, Total\_invoice\_Amount)

|  |  |  |
| --- | --- | --- |
| payment\_id | payment\_method\_id | payment\_amount |
| pmt001 | pmid001 | 1000 |
| pmt002 | pmid002 | 300 |
| pmt003 | pmid003 | 900 |

Payment (Payment Id, payment method id, Total\_payment\_Amount)

|  |  |  |
| --- | --- | --- |
| invoice\_payment\_id | invoice\_id | payment\_method\_id |
| ip001 | inv001 | pmid001 |
| ip002 | inv002 | pmid001 |
| ip003 | inv003 | pmid001 |
| ip004 | inv004 | pmid002 |
| ip005 | inv004 | pmid003 |

Invoice\_Payment (Invoice\_Payment Id, invoice\_id, payment\_id)

1. Entity Relationship Diagram

The personal auto insurance has many datapoints to be captured to calculate the most accurate premium for the named insured. The major datapoints are 1) driver information, including driving experience and age, 2) vehicle information, including the year-make-model, mileage, safety features, 3) coverage information, with both liability and physical damage coverage and selected deductibles, 4) driver’s accident/violation information.

A picture containing text

Description automatically generated

Figure B.2.1 ERD

As it can be noticed that the coverage has been split into two separate entities; since the liability coverage is named insured specific; whereas physical damage is vehicle specific. Premium is calculated taking the selected coverage and deductible limits, vehicle and driver’s information into consideration. The invoiced premium may have multiple installment payments or one-time full amount payment. Similarly, if the customer has a separate line of business with the company or a separate auto policy, the customer may want to pay once for all the invoices. Therefore, there needs an associative entity to capture this many-to-many relationship; which has been illustrated in the below ERD.

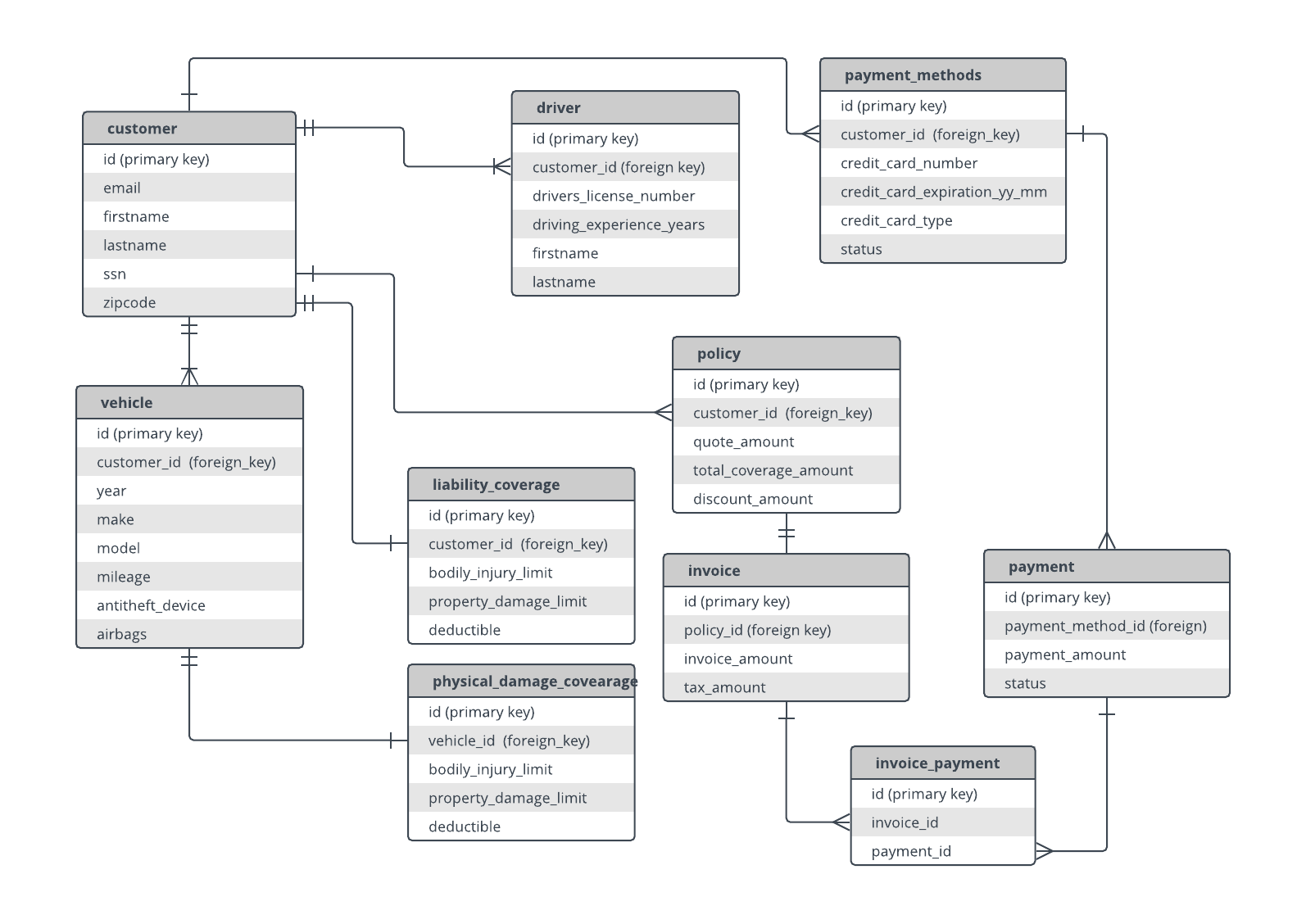


Figure B.2.2 ERD

1. Data File Storage and Access

Each of the data files in the tables will be stored non-sequentially. However, each table will have it’s own primary key (id), by which it can be identified. These primary key ids will be used as a foreign keys in other tables to have the entity relationship. The id field can be auto incremented integer field setting within the database or any unique string that is entered by the other external systems which are connected to the databases.

# System architecture

An MVC architecture consists of 1) a model – data model, 2) view – user interface 3) controller – interface between the frontend and backend. The proposed application is a perfect candidate to follow the MVC architecture.

A close up of a device

Description automatically generated

**Figure C.1 System architecture**

Figure K.1 illustrates the system architecture of the application. In an MVC system, a graphical user interface, which is the view part of the application, is the user facing part of the system. The UI communicates with the controller, which in this case is the Rating Engine (Premium Calculation Service). The controller acts as the interface between the UI and the databases. The cotroller also communicates with external systems and processes data accordingly.

External Systems:

Motoro vehicle record (MVR) – The controller makes an API call and uses the driver’s license number as a request body to receive the authenticated driving record of the customer or the driver. MVR system has both SOAP and REST APIs. Being more secured, reliable and faster response time, the REST APIs are better candidates to be used by the new application controller.

Underwriter approval process – Based on the MVR record, underwriters decide whether to approve the policy or reject a policy request. Higher the number of accidents and violations in the history, lesser the chance of getting the policy approved. This is an existing in-house system, which is being used by the internal policy admin system. This system contains a set of rules which are evaluated to make a approve/reject decision.

Billing / Payments – Existing application and can be reused by the new self-service quoting system as well. However, since a public application is going to make an API call to the billing system, the security and data privacy is an utmost priority. JSON Web Token (JWT) standard systems may be used to secure the credit card information. End-to-End data encryption is another option to secure this data both while in transit and in rest.

Forms engine – Existing application and can be reused by the new self-service quoting system as well.

# References

1. Dixit, J.B. & Kumar, R. (2007). Chapter 12 - Designing Databases. In *Structured system analysis and design*. [Books24x7 version] Retrieved from

*http://common.books24x7.com.ezproxy.umuc.edu/toc.aspx?bookid=30713*.

1. Dennis, A., Wixom, B.H., and Tegarden, D. (2015). System Analysis and Design, Fifth Edition, John Wiley & Sons. Retrieved from *https://learning.oreilly.com/library/view/system-analysis-and/9781118057629/18\_chap09.html#chap09*
2. Mak, Gary. (2008). Chapter 10 - Spring MVC Framework. In *Spring recipes: a problem-solution approach*. [Books24x7 version] Available from *http://common.books24x7.com.ezproxy.umuc.edu/toc.aspx?bookid=29010*.