RfQ Response

Q1. Describe the architecture of a typical Rails application.

Rails typically uses an MVC architecture, which stands for, Model, View, Controller respectively.

A brief explanation of each part of the architecture follows below:

Model:

The model is the storage of our data and logic.
In Rails the model represents the information in the database as well as the logic behind any validation.
The model interacts with the database anytime we want to amend our data, often using the CRUD system(create, read, update, delete)

View:

The view is what's sent back to the browser, the GUI.
The view is the front-end of the application.

The view is the part of the architecture which translates our data into a human readable format. The view communicates with the controller after it has made a request for data objects from the model. The view also communicated back to the controller when necessary to make changes to the model, for instance, if we had a model called User, we might have a view which allows us to edit the a user, in this view, we would input our changes, then our changes would go to the controller, which would then go to the model to be saved/updated.

In Rails the view contains html.erb files, where the embedded ruby is fairly simple, mainly used to display data from the model, or used to create fields to edit/add to the model.

Controller:

The controller is what's needed to manage the view and model of our application. The controller is the part of the MVC architecture, where we directly communicate with the model, and the view. The controller will often recieve data from the model to be used in the view, and will also take inputs from the view to update the model.

Q2. Identify a database management system commonly used in web applications and discuss the pros and cons of this database.

PostgreSQL:

“PostgreSQL is a powerful, open source object-relation database sytem that uses and extends the SQL language” - postgresql.org

PostgreSQL dates back to 1986, when it was worked on as part of a project at the University of California at Berkeley, called POSTGRES, and since then it has had 30 years of active development on the core platform.

Pros:

- *Extensibility:*
  PostgreSQL isn’t a fixed database system, if you’re in need of an additional feature in Postgres you can add it, which is relatively difficult with other database systems.
- *Security:* Postgres has inbuilt features which relate to security (SHA-256, GSSAPI, SSPI, etc.), and it also has extensions available which you are able to use to enhance the security of your database.

Cons:

- *Open source:* Being an open source database application, PostgreSQL is not owned by a singular organisation. Given this fact, the system comes with no warranty and it has no liability.
- *Slower performance:* PostgreSQL is known for having a number of issues with performance and backup recovery.
  PostgreSQL due to it’s relational database structure, can struggle and perform slowly when working with a large number stores located in rows and columns of a table containing many fields of inforation

Reference 1
Reference 2

Q3. Discuss the implementation of Agile project management methodology.

The Agile methodology is an alternative project management system, that focuses on separating a project into smaller parts, with an aim to help prioritize customer feedback in the development process.
Once work begins on a project, teams will cycle through a process of planning, executing and evaluating each segment, all the while being in near constant contact with customer’s to get their feedback to ensure requirements are being met.

The Agile methodology is good because instead ‘betting everything on a “big bang” launch’, Agile allows a team to deliver their work in small, consumable increments which can be reviewed, and have feedback come in, regularly.

Often the Agile methodology will work as such:

1. Making a roadmap for an application, which will outline timelines and goals for the future of the project

2. Plan and implement sprints, which are our smaller goals, which are generally planned to roughly take the same amount of time.
   - During these sprints, standup meetings may occur daily or weekly to ensure everyone is up to date on what is going on in the project, and to ensure nothing is being left behind or taking too long
3. Reviewing how each sprint performed, and if there were any lacking areas that need to be assessed for the next sprint.

Reference 1
Reference 2

Q4. Provide an overview and description of a standard source control workflow.

Feature branch workflow:

A method of source control is important in any project, as it’s arguably the best way to protect a project from any difficulty or risk when working on a collective codebase. Source control is essentially a way for multiple developers(or just the one) to work together on a project without causing a mass amount of conflicts within the codebase, and without simply overriding someone else’s work. It also gives a company or developer a safe place to track and keep a working version of there application online.

The feature branch workflow is a source control style which has the core idea that each feature being developed should take place on a dedicated branch, instead of being worked on the master branch.
Doing this makes it easy for a number of developers to work on a feature together without harming the main codebase, which also means the master branch will never be broken.

Essentially the way this style would be implemented is as such:

1. A feature will be planned and discussed
2. A new branch will be created for this feature, generally forking from the master branch
3. Developers will work on this new branch at all times when working on the new feature
4. When the feature has been completed and tested, it will be merged into the master branch

reference

Q5. Provide an overview and description of a standard software testing process.

Test driven development:

Test driven development is an approach in which test cases of a program are developed before the actual program is developed, to specify and validate what the code needs to and will do.
TDD, starts with designing and developing test for every functionality of an application, then moves onto coding only if said test has failed, which avoids duplication of code.

A brief example of how this is implemented is as such:

1. a test is written which defines how a part of the program should behave
2. the test is run, to evaluate whether or not the program is behaving as planned
3. if the test fails (and only if it fails), code is written to fulfil the testing requirements
4. tests are run again, and code is refactored if necessary
5. rinse and repeat

reference

Q6. Discuss and analyse requirements related to information system security and how they relate to the project.

As the planned project is a marketplace application, there are a number of security measures that are required to ensure we meet ethical and legal obligations.

As a marketplace application, we will be handling a lot of user data, which needs to be protected, so our user's aren't at risk of having their data sold, or used maliciously by unknown peoples.

Firstly we'll have to consider our requirements when taking in user data. As a marketplace, generally we'll be dealing with user to user relations, and as such, we need to ensure that we will be storing a lot of data that is unique to each unique user. The data we will be storing, needs to be stored in a way that protects it from potential attackers and other users, as it could contain very personal information, such as phone numbers/addresses/full names/ etc.

Another big thing we'll have to think about is payment, obviously the easiest way for us to deal with this would be to put the onus on the users to discuss and agree to payment terms, but, if we decided to use some payment methods in our application, we would need to think about the legal and ethical requirements put upon us when we decide to do so. We would need to ensure we're following PCI standards, which are the rules and regulations which determine how payment data is stored online.

Other things we'll need to worry about would include things such as illegal items being sold on our site, as well as scammers. We would need a way to ensure that items being sold on our site meet legal requirements, and are pertinent to our values and plan for this application. We would need to ensure our user's safety by implementing ways to filter out scammers, on the buying and selling side of our application.

reference

Q7. Discuss common methods of protecting information and data and how you would apply them to the project

Encryption

As our application will certainly contain very sensitive information to our Users, we need to ensure that we're taking appropriate measures to protect their data. A very commonly used way to protect data online is through encryption.

Encryption is a method of protecting one's data by encoding it, using some sort of encryption protocol, such as AES, RSA, Triple DES etc. The encrypted data can only be accessed and decrypted by someone who is in possession of the correct encryption key.

The encyrpted data to an outside entity who isn't in possession of the correct key, looks like garbage, and is completely useless, which is why it's a very good way of protected the data of our users, especially when we come to talk about highly sensitive information such as personally revealing data, and credit/debit card numbers.

In our project, we should be encrypting our data before we save it into our database. Our less important data doesn't necessarily need to be encrypted, but for our highly sensitive data, such as important user data(credit card numbers, phone numbers, address), we would aim to encrypt it before putting it into our database, and depending on the software used, the database itself would itself be encrypted as well.

Q8. Research what your legal obligations are in relation to handling user data and how they can be met for the project

If the ACME corp has an annual turnover of at least $3M, they are required by law to comply with the Privary Act of 1988.
The Privacy Act includes 13 privacy principles which apply to some private sector organisations, as well as most Australian government agencies.

Even if ACME corp isn't required by law to follow the Privacy Act, it would be smart to follow it anyway, as it would keep us safe from potential breaches of privacy, and will instill a great amount of trust with users

To comply with the Privacy Act, the ACME corp would be required to: - Tell users why their personal information is being collected, how it will be used, and who it will be disclosed to - Give users the option to remain anonymous, through the use of a pseudonym or simply by not identifying one's self - Give the users an option to access their own personal information - Give users the ability to stop receiving unwanted direct marketing - And give users the ability to make a complaint about the corp, if they think we've mishandled their personal information.

https://www.oaic.gov.au/privacy/the-privacy-act/

Q9. Describe the structural aspects of the relational database model. Your description should include information about the structure in which data is stored and how relations are represented in that structure.

Data in a relational database is often organized into tables. These tables will have rows and columns which represent the attributes and records of the table. When working with data that requires relationships between tables, this can be done by using foreign keys within a table. For example.
If we were to have a book table, and an author table, a book can have many authors, and an author can have many books, so we would have both tables, and a joiner table, which will take both primary keys from each table to join them into a many to many relationship.

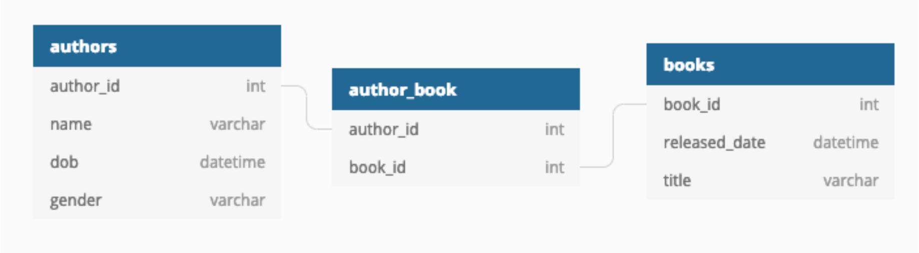


Figure 1: many to many relationship image

Relation types we can have are as follows:

- one to many
- one to one
- many to many

To conclude, the structure is as follows:

- Data is stored into tables, with columns(data types), and rows(records)

- Tables contain primary keys, which can be used in other tables to display relationships between the tables (foreign keys)
- These relationships can be one of three types, many to many, one to many, or one to one

Q10. Describe the integrity aspects of the relational database model. Your description should include information about the types of data integrity and how they can be enforced in a relational database.

The integrity of a database is enforced by using a series of constraints and rules. Generally the three types of integrity constraints are as follows:

- Entity integrity
  Entity integrity is the idea of having a primary key for each table. This rule establishes the table integrity by providing a primary key which is required, and this key must be placed on a column that is unique, and not null.

- Referential integrity
  Referential integrity is the concept of keeping a foreign key for relationships between tables. The foreign key generally refers to a primary key in another table which we are forming a relationship to, but there can be times where the foreign can be null, meaning there is no relationship, or the relationship is unknown.

- Domain integrity
  Domain integrity refers to constraints that enforce the rule that all entries in a column of a table are only single entries, that are of the matching data type. The domain for the column also ensures that entries match requirements/rules of the attribute, for example, if we have a column that is in date format, we shouldn't be allowing names.

https://en.wikipedia.org/wiki/Data_integrity#Integrity_types

Q11. Describe the manipulative aspects of the relational database model. Your description should include information about the ways in which data is manipulated (added, removed, changed, and retrieved) in a relational database.

In the relational database model, there are a number of ways to manipulate the data in our databases. The specific functions to manipulate the data can differ from each database system, but the four main functions to modify our data are as follows:
- Insert - Is used to insert an entry into a table or relation
- Delete - Is used to remove an entry from a table or relation

- Modify - Is used to edit values from an existing table or relation
- Select - Is used to choose an entry via query or indexing

https://www.guru99.com/relational-data-model-dbms.html#4

Q12. Conduct research into a marketplace website (app) and answer the following parts:

Ebay

- List and describe the software used by the app.
  - `JavaScript`
    JavaScript is a text-based programming language, that is widely used, and arguably the most use language for the web, in the world. It is used both on server and client side, and it allows you to make interactive web pages. There is a lot that JavaScript can do, and it's one of the 3 main things necessary in web development, those being CSS, HTML, and JavaScript.
    Things JavaScript can do include
    - Create dynamically updating content
    - Control multimedia (animating images, playing videos, etc.)
    - Develop interactive pieces of web pages
  - `Node.js`
    Node.js is an asynchronous event-driven JavaScript runtime environment, that was designed to build scalable real-time network applications.

  - `Java`
    Java is a high level, class based, object-oriented programming language with a similar syntax to C++. Java is used widely, and can be used to create single app applications, as well as applications that will be used across servers.
    Java was intened to let developers 'write once, run anywhere' meaning, that once Java code is compiled, it is able to be run on all platforms that support Java without the need to recomple said code.
  - `ES6`
    ECMAScript (ES6) is a general purpose programming language. It's a standard for the JavaScript language meant to ensure the interoperability of we pages across different web browsers.
  - `Apache Tomcat`
    Apache Tomcat is a free, open-source implementation of a number of technologies, including, Java Servlet, JavaServer Pages, Java Expression Language, and WebSocket. A Java Servlet is a software that enables a server to handle dynamic java-based content using the HTTP protocol. This along with all the other Java based technologies together, gives developer's a 'pure Java' web server environment to

run applications built with Java.

  - `Cassandra`

    Cassandra is a free, open source, wide column store (uses tables, rows, and columns but names and format of columns can vary from row to row in the same table) NoSQL database system, designed to handle large amounts of data across many servers.
  - `Oracle`

    Oracle is a database commonly used for running online transaction processing, data warehousing and mixed database workloads.
  - `Marko`

    MarkoHTML is a 'reimagined' language for building dynamic user interfaces. Marko is very similar to standard HTML, but it extends the language to allow a developer to build modern applications in a declarative way.

- Describe the hardware used to host the app.

  Ebay hosts it's website on an OpenStack private cloud platform that they developed themselves.

  OpenStack is a cloud operating system that manages and provisions all it's networking, compute, and storage resources by the use of common API euthentication mechanisms. OpenStack has a number of benefits which has made it a widely used technology, and they include:
  - Great industry support
  - Compatibility
  - Scalability
  - Security

  OpenStack also provides an easy to manage panel that provides visibility, control and easy access to power management tools, which makes it easy for usetrs to manage/monitor their services.
- Describe the interaction of technologies within the app

  Users will interact directly with the application through the front-end via their browser of choice, which is run and hosted on Ebay's private cloud platform. The user's browsers will send http requests to the server, and in turn Ebay's servers will run through their extensive databases and models to find an answer to that request which will then be returned to the user.

  The front-end of Ebay would be controlled and managed by JavaScript (inc. ES6), and Node.js.

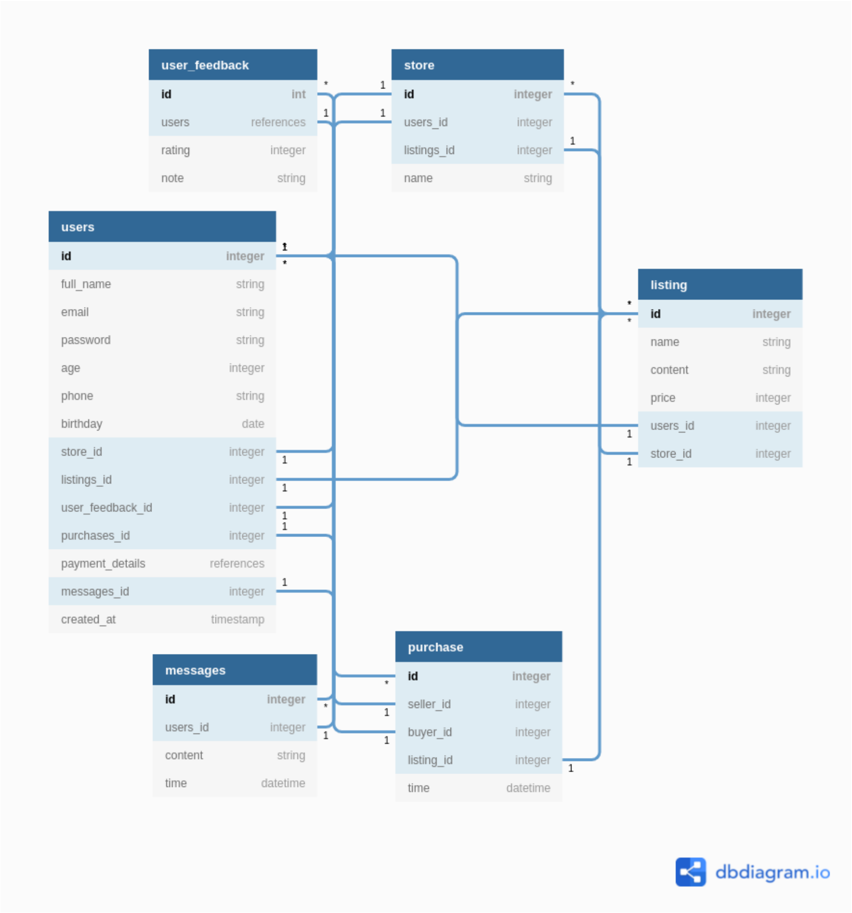
  Node.js is used to create and build fast scalable applications, which is used alongside Marko for the user interface.

  There is a lot that goes on, in the backend of Ebay. The main database system used for ebay is Oracle, as a large majority of the use of Ebay revolves around making online purchases and processing transactions.

  Apache TomCat is the software which is running the servers, it being a piece of software which combines a number of Java-based technologies to run applications built with Java.
- Describe the way data is structured within the app

  We can assume that the data held in Ebay's servers are structured in a relational database. Each object in the database will be stored with column and row attributes that represent each part of the object. Each table in the database, when necessary, will have foreign keys that will point to other tables to create relationships between the tables. This structure builds a solid foundation for a two-way marketplace application.

- Identify entities which must be tracked by the app
  - Users: tracks all information relevant to the users, could include, usernames, names, passwords, emails, addresses, etc.
  - User Feedback: a rating from another user, which contains a star value and short note.
  - Listing: tracks all information relevant to something that's being listed
  - Stores: tracks all items in a user's specific store
  - Messages: tracks the message and both user's in the interaction
  - Purchase: tracks a specific purchase from a user
  - Payment details: tracks a user's payment details
- Identify the relationships and associations between the entities you have identified in part (e)
  - Users:
    - Has many: user feedback
    - Has many: listings
    - Has one: store
    - Has many: messages
    - Has many: purchases
    - Has one: payment details
  - User Feedback:
    - Has many: users
  - Listing:
    - Has one: users
    - Has one: store
  - Stores:
    - Has one: user
    - Has many: listings
  - Messages:
    - Has many: users
  - Purchase:
    - Has many: users
    - Has one: listing
  - Payment details:
    - Has one: user
- Design a schema using an Entity Relationship Diagram (ERD) appropriate for the database of this website (assuming a relational database model)



user_feedback	
id	int
users	references
rating	integer
note	string

store	
id	integer
users_id	integer
listings_id	integer
name	string

users	
id	integer
full_name	string
email	string
password	string
age	integer
phone	string
birthday	date
store_id	integer
listings_id	integer
user_feedback_id	integer
purchases_id	integer
payment_details	references
messages_id	integer
created_at	timestamp

listing	
id	integer
name	string
content	string
price	integer
users_id	integer
store_id	integer

messages	
id	integer
users_id	integer
content	string
time	datetime

purchase	
id	integer
seller_id	integer
buyer_id	integer
listing_id	integer
time	datetime

References:
- https://stackshare.io/ebay/ebay
- https://nodejs.org/en/about/
- https://en.wikipedia.org/wiki/Apache_Cassandra
- http://tomcat.apache.org/
- https://vexxhost.com/blog/benefits-openstack-public-cloud-enterprises/