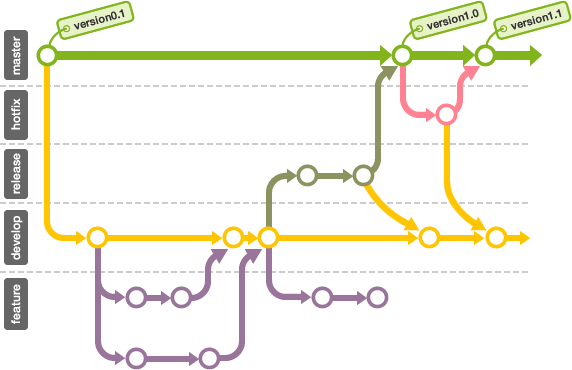
## Git

Git is a tool that is used in companies where there are a lot of people who work on the same project. It has version controlling and changes can be rolled back. Companies uses tools like [BitBucket](https://bitbucket.org/) as a repository, but for this project, we can also use [GitHub](https://github.com/) which is better known. Only problem is that with GitHub, all the code produced there will become public domain (except of you pay the subscription).

Git has a unique feature for development. It has a branch system. In the beginning, there is a master branch, which acts like the production version of the project. New branches can be created from the master branch to make changes which will not affect the master project. This will act like the development branch. Here is an example of how the branching works:



The most daunting part of this is the Pull, Commit and Push mechanism of Git. Once you get used to it, it is very useful. In the beginning, everyone involved in the project must Clone the project from the remote repository on GitHub or BitBucket. This basically does the same thing as creating files on your computer in the same fashion as DropBox. The only difference is that any changes to the cloned files needs to be committed and then pushed before it can be accessed by other people in the project.

The Pull feature is used to pull the latest version from the remote repository to your local repository. This is necessary to do before you commit your change to make sure that your change incorporate other peoples changes as well.

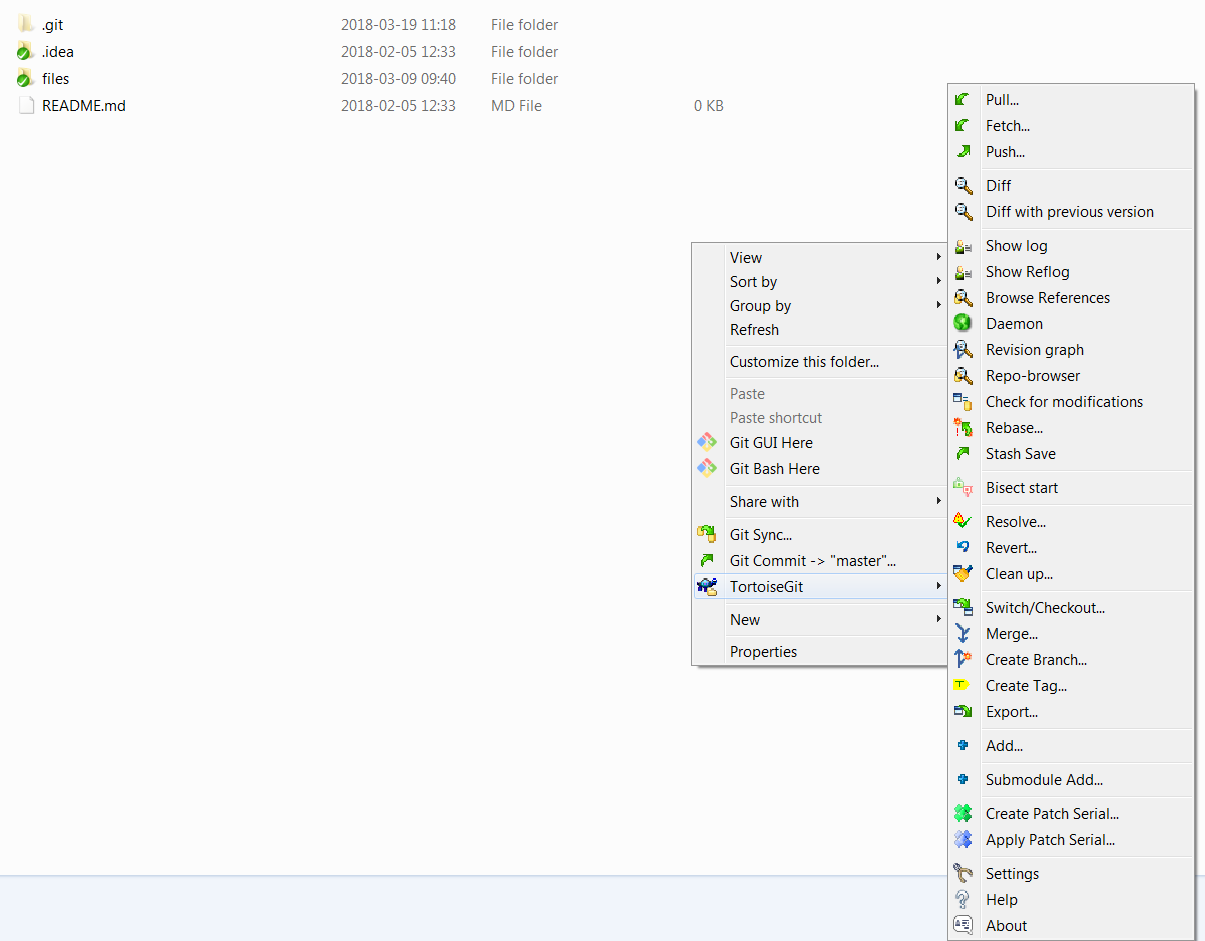
The commit is where you put your changes into your local repository and create the packaged to be pushed. This will also create a log on what you did in the project as well.

The Push part is where you move the code from your local repository to the remote repository where people can also pull your changes. If someone made a change in the same file than you, it will tell you that a merge is necessary before it can be pushed. That is why you can never overwrite someone’s changes without your knowledge.

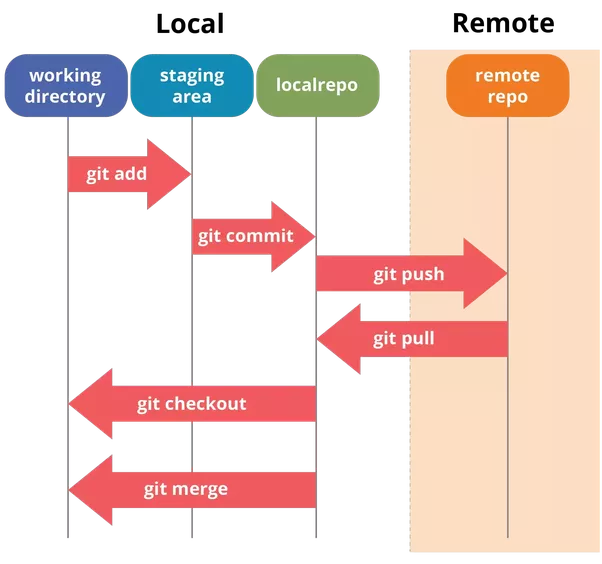
With the merge, it will give you a screen with 2 windows. One window has your changes on your local repository and the other is the copy on the remote repository. You can then decide if incorporate your changes into the file from there, effectively merging your changes with the file on the server. From here you can decide if you want to overwrite someone’s code.

To make this effective we need to use a IDE which is Git enabled. A compatible IDE will add the file to your storage area for you without you having to do it manually. You also will need a Shell interface like [TortoiseGit](https://tortoisegit.org/) or [SourceTree](https://www.sourcetreeapp.com/) that acts like “DropBox” to push, commit and pull changes to your computer, tell you which branch your working on, and also the log of changes to the project.

Here is just an example how TortoiseGit works in a folder that is linked to a project:



Here is a nice representation of how the Push, Commit and Pull works:



I will be able to provide training for anyone who does not know how to use this or if you struggle to come right with this. It is a very useful tool and would highly recommend this.

## Credit Card Structure.

I can give you a summarized version of this structure. I can give you some insight of how a bank generally built their system. I would recommend that we build the same kind of structure and as soon as the foundation is done, we should look at getting the primary objective sorted out first before looking at adding some extra features.

On most programs, we build it according to the [MVC principle](https://www.tutorialspoint.com/design_pattern/mvc_pattern.htm) (Model/Database Layer, View/Presentation Layer and Controller/Business Logic layer). This prevents exposure of database assets to the front end. This principle also makes up the framework on creating a proper [API](https://docs.microsoft.com/en-us/aspnet/web-api/overview/getting-started-with-aspnet-web-api/tutorial-your-first-web-api). Creating API’s make it possible to reuse features inside the project, like for example getting Card Information.

### 2.1 Now for the Credit Card Part.

All cards have an account which it is associated to. The account has a user associated with it as well. Under that account, a person can have multiple cards. One needs to be the Primary card and the others needs to the secondary cards.

Transactions can be viewed on an account level or on a plastic level.

### 2.2 Plastic/Card Record Layout

The following table is the common structure of a Card in the bank. The green fields are NOT on the record on a bank’s database, but stored more securely. Passwords, CVV and Expiry date are never stored in a database, but rather hardware stored. For the purpose of this project, I would recommend storing that data on the database under the card record to keep it simple.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Field** | **Size** | **Description** |
| PK | Plastic Number | 23 | The plastic is generally a 16 character field with the rest being leading zeroes for example 00000004512368512264512.   The first 3 numbers are registered to the bank by either Visa or Mastercard. In the example above it will be 451.  The next 3 identifies the product, so those 3 numbers must correspond with the product field above. In the example above it will be 236.  As you can see, the last 3 digits (512) of the plastic number identifies the card in the account range.  The last digit is a checksum digit. |
| FK | Company ID | 03 | Generally used to distinguish which division in a bank the card belongs to. An example is having a branch in RSA and one in Bot. Both countries will have different codes. We can make up our own. If we only use RSA, we do not need this field. |
| FK | Product | 03 | Used for main product classifications like General (GEN), Corporate (CRP) and Business (BUS) |
| FK | Sub-Product | 03 | Generally used to indicate which product house this card belongs to PLA for a platinum card for example or GLD for gold card. |
| FK | Account Number | 23 | Account number linked to this plastic generally it will look like this: 00000004512368512264000. The last 3 digits are generally zeroes. This is to make space for cards. |
|  | Account Type | 01 | Primary Card (P) or Secondary (S). An account must have a primary before a secondary card can be opened. |
|  | Verified by Visa Indicator | 01 | Active (A), Inactive (I). This is used for if a card has been activated for Visa verification on online purchases. |
|  | Primary Plastic Number | 23 | The primary plastic number which the secondary card is attached to. This number must only be populated if this record is a secondary plastic. |
|  | Current Status | 03 | This is the status of the card at this moment. If it is inactive, open or deleted. |
|  | Previous Status | 03 | This is what the current status was before it is changed. This is to make sure that the card statuses do not get messed up. For example, you cannot have an expired/delete previous status, but the current status is open. |
|  | Last Date Status Change | 08 | The last time the status of the card was changed |
|  | Emboss Line 1 | 40 | Emboss line 1 is generally empty and only used for business cards to add the company name to the card. |
|  | Emboss Line 2 | 40 | Emboss line 2 will always hold the name of the person to whom the card has been issued to. |
|  | CVV Number | 03 | This is the code used for online purchases. |
|  | Last Withdrawal Limit Change | 08 | The last time the credit limit was changed |
|  | Card Created | 08 | The date the card was created |
|  | Card Opened |  | The date the card was activated/have been opened. |
|  | Card Expiry date | 06 | When will the card expire (format mm/yyyy). On the card the format Is only mm/yy |
|  | Password | 20 | The password of this plastic. |
|  | Reissued Card Number | 23 | If this card is replaced, this field must be populated with the next card. Otherwise this field is empty. |
|  | Reason | 40 | If plastic was closed/cancelled or deleted, this field must contain the reason for the action. |

### 2.3 Account Record Layout

This record layout is the layout of an account. There is a lot of data, but you can decide what we need and what we can remove. If you do not understand a field, feel free to ask.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Field** | **Size** | **Description** |
| PK | Account Number | 23 | The plastic is generally a 16 character field with the rest being leading zeroes for example 00000004512368512264512.   The first 3 numbers are registered to the bank by either Visa or Mastercard. In the example above it will be 451.  The next 3 identifies the product, so those 3 numbers must correspond with the product field above. In the example above it will be 236.  The last digit is a checksum digit. |
| FK | Company ID | 03 | Generally used to distinguish which division in a bank the card belongs to. An example is having a branch in RSA and one in Bot. Both countries will have different codes. We can make up our own. If we only use RSA, we do not need this field. |
| FK | Account Product | 03 | Used for main product classifications like General (GEN), Corporate (CRP) and Business (BUS) |
|  | Account Type | 04 | Cheque (Cheq), Savings (Save), Transmission (Tran), Bond, Subscription Share (Sub) |
|  | Branch Code | 06 | Branch code of the branch where the account was opened |
|  | Inactive Cards | 03 | Number of inactive cards on account. |
|  | Open Cards | 03 | Number of open/active cards on the account |
|  | Closed Cards | 03 | Number of closed/canceled/destroyed cards on this account. |
|  | Primary Plastic Number | 23 | The primary plastic number associated to this account. |
|  | Current Status | 03 | This is the status of the card at this moment. If it is inactive, open or deleted. |
|  | Previous Status | 03 | This is what the current status was before it is changed. This is to make sure that the card statuses do not get messed up. For example, you cannot have an expired/delete previous status, but the current status is open. |
|  | Last Update Date | 08 | Last date the account was updated |
|  | Last Update Time | 06 | Last time the account was updated |
|  | Last Update Operator | 08 | The last Operator ID that updated the account. This can be null if the terminal is online. |
|  | Last Update Terminal | 08 | The last terminal that updated the account. This can be an online update or an update done in a bank branch |
|  | Account Open Date | 08 | The date the account was opened |
|  | Account Close Date | 08 | The date the account was closed |
|  | Cycle Date | 08 | The cycle date of the account. When statements must be done and fees must be paid. |
|  | Account Expiry Date | 08 | The date the account will expire. |
|  | Initial Issuing Date | 08 | The date which the first credit card was issued on this account. |
|  | First Activity Date | 08 | The date which the first transaction was done. |
|  | Last Known Activity Date | 08 | The last date any action was done on this account. |
|  | Last Reissuing Date | 08 | The last date a card was reissued on this account. |
|  | Last Fraud Activity Date | 08 | The last date fraud was detected on the account |
|  | Last ATM Activity Date | 08 | The last time an ATM was used for this account |
|  | Last Auth Required Date | 08 | The last date an authorization was required on this account. |
|  | Last Auth Approval Date | 08 | The last date an authorization was approved on this account. |
|  | Outstanding Auth Amount | 11.2 | The total amount of auths that are outstanding |
|  | Outstanding Auth Count | 07 | The total number of auths that are outtstainding |
|  | Outstanding Extended Credit Auths | 07 | The total number of auths outstanding on budget transations |
|  | Auths Approved | 03 | The total number of auths approved |
|  | Auths Declined | 03 | The total number of auths declined |
|  | Last Batch Process Date | 08 | The last date this account was processed via batch |
|  | Last Transaction Date | 08 | The last date there was a transaction done on this account |
|  | Last Transaction Amount | 07 | The last transaction amount done on this account |
|  | Total Transactions | 03 | Total transactions done over this account |
|  | Reason | 40 | If account was closed, this field must contain the reason for the action. |

### 2.4 Transaction Record Layout

This is the record that holds the transaction data for a plastic.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Field** | **Size** | **Description** |
| PK | Auth Code | 05 | The Auth code that gets passed by a Point of Sale Device it is used in the place of the Transaction Code if there are no transaction code. |
| PK | Transaction Code | 05 | The transaction code that gets passed by a Point of Sale Device |
| FK | Company ID | 03 | Generally used to distinguish which division in a bank the card belongs to. An example is having a branch in RSA and one in Bot. Both countries will have different codes. We can make up our own. If we only use RSA, we do not need this field. |
| FK | Account Product Code | 03 | Product code of the account. Used as part of the key. |
| FK | Account Number | 23 | Account number on which the transaction was done on. |
| FK | Plastic Number | 23 | Plastic number on which the transaction was done on. |
|  | Transaction Status Indicator | 01 | Typical Status indicators = Information Only (I), Normal Item (N), Disputed Item (D) |
|  | Transaction Source Indicator | 01 | MasterCard (M), Visa (V), Local Bank (L), Competitor Local Bank (C) |
|  | Transaction Source | 05 | Merchant Code |
|  | Source Currency Code | 03 | The currency in which the transaction was done from |
|  | Destination Currency Code | 03 | The currency in which the transaction must process. Generally in our case it is in Rand (ZAR) |
|  | Transaction Date | 08 | Date on which the transaction was done. |
|  | Transaction Time | 06 | Time the transaction was done. |
|  | Original Posting Date | 08 | The date which the transaction was posed (processed) from an Auth |
|  | Transaction Type | 03 | Payment/Settlement (PU), Auth (AU), Auth Reversal (AUR) |
|  | Transaction Amount | 11.2 | The amount of the transaction |
|  | Adjustment Reason Code | 03 | The code for making an adjustment to the amount. Can only be done from the Transaction source. |

### 2.5 Customer Record Layout

The user information is not in my field of work. Our Customer Information Systems division holds this data. I will try my best to provide you with the most detailed information on the customer information required.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Field** | **Size** | **Description** |
| PK | Unique Customer Number | 10 | This is a unique number generated for every customer |
|  | Title | 04 | Title of the customer |
|  | Name | 40 | Name of the customer |
|  | Surname | 40 | Surname of the customer |
|  | Gender | 01 | Male (M), Female (F) |
|  | Nationality | 01 | Foreign (F), Local (L) |
|  | Date of Birth | 08 | Customer’s date of birth |
|  | ID | 13 | This is the id number of the customer. It is only applicable if the customer is a South African national |
|  | Passport Number | 15 | This is the passport of the customer. It is only applicable if the customer is a foreigner. |
|  | Cellphone Number | 13 | The cellphone number of the customer. |
|  | Home Number | 13 | The home number of the customer. |
|  | Work Number | 13 | The work number of the customer. |
|  | Fax Number | 13 | The fax number of the customer. |
|  | Email Address | 25 | Email address of the customer. |
|  | Preferred Contact Method | 01 | Email (E), Cellphone (C), Home phone (H), Work phone (W), Fax (F), Post (P) |
|  | Company or Business Name | 40 | Only applicable if this is a company or business user. |
|  | Address Line 1 | 40 | Address Line 1 |
|  | Address Line 2 | 40 | Address Line 2 |
|  | Address Suburb | 40 | Suburb |
|  | Address City | 25 | City |
|  | Address Province | 40 | Province |
|  | Address Postal Code | 04 | Postal Code |
|  | KYC Indicator | 01 | Verified (V), Unverified (U), Expired (E). KYC stands for Know your Customer. It is the documents which the bank asks for, like the Proof of Residence and the Certified ID for example. This is set manually in the branch. |

## Rules

Here are a few constraints we have for our system:

1. By law, the bank cannot block a transaction even if it is detected as fraud. Only a customer can block it, so our task will be to let a customer know if we detect any fraudulent transactions.
2. All international transactions can only be received though Visa or Mastercard
3. To mimic a point of sale transaction from another country, we can create an app that can mimic it. I would also recommend creating an app where the customer can specify if they go out of the country and for how long they will be out.

That is all the rules I can remember.