**ABOUT**

Xenon Health is currently using Health Fusion for its billing and claim purposes.

The system is easy to use but is proving to be costly. Having an in-house billing system will prove cost effective in the longer run.

Building such a system:

Any system that deals with patient information needs to comply with HIPAA privacy rules.

These rules specify the way the patient information should be protected by specifying the guidelines that need to be followed while designing such a system.

The guidelines state what needs to be done but do not provide any help on how to do them. Creating an audit log page is a requirement of HIPAA, which has been completed.

**Work so far**

After you open the BillingApp directory you will notice that the whole thing is divided into multiple apps, namely: accounting, accounts, base, BillingSystem, claims, dashboard, displayContent, infoGatherer, media and report.

Accounting: This takes care of accounting of claims. It includes models like claims, payments, apply and create.

Accounts: This take care of the login system for users.

Base: A base model to be used in other models

BillingSystem: The main app which contains setting.py

Claims: Handles claim creation and generation.

Dashboard: Includes everything associated with the dashboard (main) page.

DisplayContent: This takes care of viewing of patients, going back to claim history of patients etc. Mimics the healthFusion website in functionality.

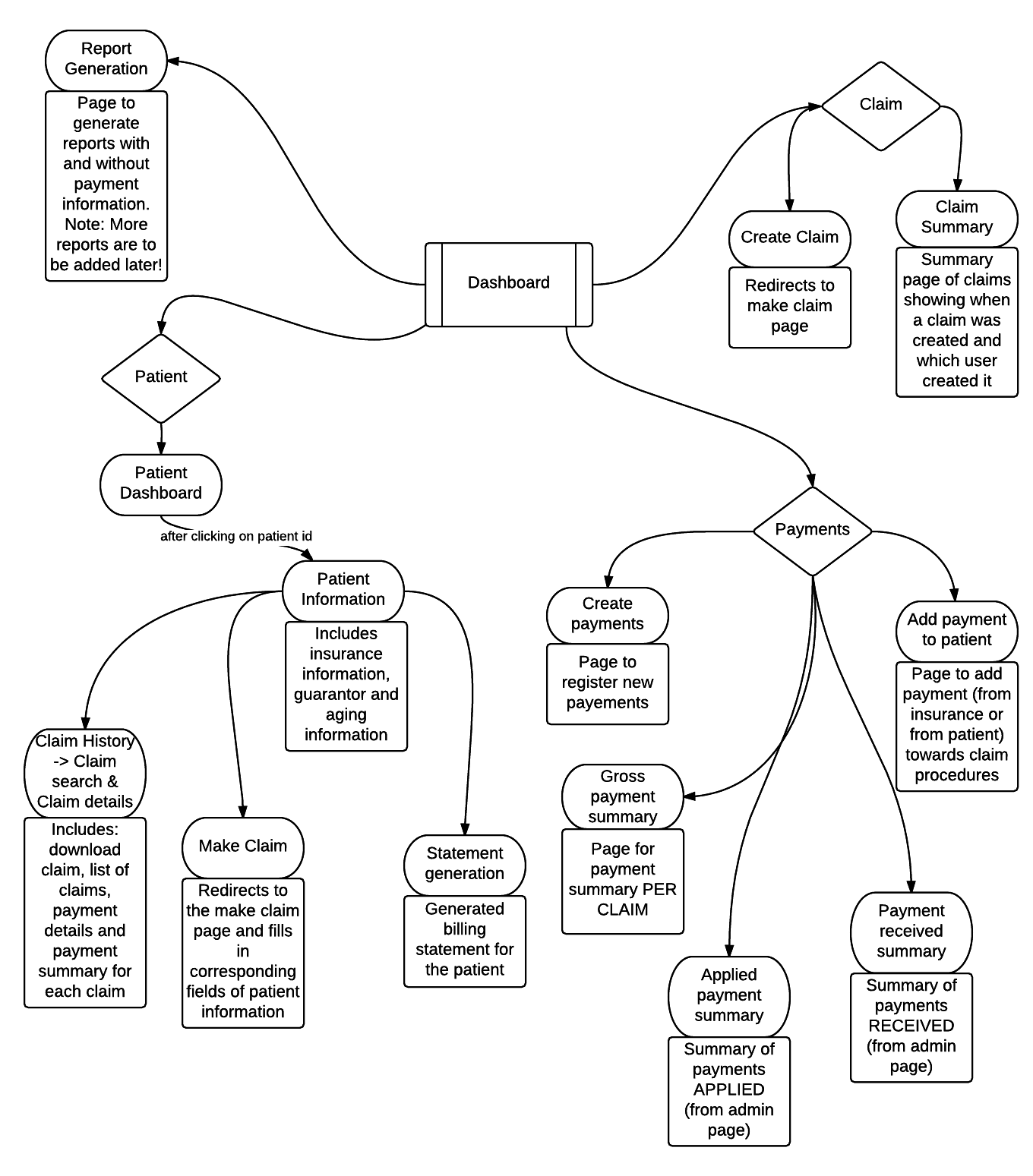
infoGatherer: Contains models for the database like personal information and diagnostic code. It also contains views.py for claim generation.

Media: Stores all generated claim and statements. These can viewed or downloaded from the app, in the browser.

Report: Contains stuff relating to report generation, like transaction reports.

**Flow of the app:**

The app starts with the dashboard, which is connected to all other components in the app. There are links in the navbar too, which is not covered here. There are links on the dashboard for guidance to every part of the app (directly or indirectly).



**Audit log (info/auditlog):**

Maintaining an audit log is a HIPAA requirement, which has been completed. This log maintained entries of old and new values for any change made to the database via the admin page. It also has the ability to see who made the changes and when were the changes made. The whole thing is split into three parts for EACH table: modified (keeps track of changes made to existing entry), created (keeps track of new entries) and deleted (keeps track of deleted entries).

**Make Claim (info/postad):**

This form is used to generate claims. The generated claim opens in a new tab as a pdf. This claim generation also triggers things like saving the claim in the server and saving the path to the saved claim in the database, creating claim object, creating Procedure objects, creating Charge objects. These saved claims can be retrieved from the claim history page of a patient.

**Admin Page (/admin/):**

The billing system is split into two parts:

One is the front end that is accessible to the whole of billing team (through valid logins) that will allow them to add patient, guarantor, insurance and claim information.

The other is a backend – admin panel that will allow specific users to add, change and deleted existing records in all displayed models.

**IMPORTANT COMMANDS**

To run the project:

python manage.py runserver

To view the project:

<http://127.0.0.1:8000/dashboard/>

To make changes to the database:

1. python manage.py makemigrations
2. python manage.py migrate

**LOGINS**

Django Admin:

Username: admin

Password: Xenonhealth

MYSQL:

Host: localhost

Port: 3306

Name: xenonhealth

User: root

Password: Xenonhealth

Login for health fusion:

User: xenonga

Pass: xenon856$

**INSTALLATIONS**

Python:

<https://www.python.org/ftp/python/2.7.11/python-2.7.11.msi>

Test Setup:

1. Upon completion of installation, open a command prompt and type python. If you see the Python prompt, installation was successful. If not you will have to set your Windows installation’s **PATH** environment variable by adding the line “C:\python27;C:\python27\scripts;” to the already existing path variable.
2. Setup **PYTHONPATH** if not already set to “C:\Python27\Lib\site-packages\”

Installing Setuptools and PIP

Requirements:

The requirements for this project are in requirements.txt file in the root directory. Use this to install all requirements (this file includes Django and mysql).

pip install -r requirements.txt

Solving visual c error of My SQL:

Goto: http://www.codegood.com/archives/129

And run: MySQL-python-1.2.3.win32-py2.7.exe

Don’t forget to migrate before running the server for the first time.

**REFERENCES**

Django:

<http://www.tangowithdjango.com/book17/>

http://www.djangobook.com/en/2.0/index.html#

Excel Data Access and Manipulation:

<http://pyexcel.readthedocs.org/en/latest/>

HIPAA Documentations:

[HIPAA Security Series #4 - Technical Safeguards - techsafeguards.pdf](http://www.hhs.gov/ocr/privacy/hipaa/administrative/securityrule/techsafeguards.pdf)

[HIPAA Security Series #2 - Administrative Safeguards - adminsafeguards.pdf](http://www.hhs.gov/ocr/privacy/hipaa/administrative/securityrule/adminsafeguards.pdf)