Mae Tao Clinic Laboratory

LAboratory Test Results System: LATRS

System architecture documentation. (v1.0)

Contents

System background and design

LATRS was developed to store medical laboratory test result data. Requests for medical **tests** are initiated from **departments**.

Tests may have a **patient** attached (in the case of HIV, they don't, due to privacy concerns).

Tests are undertaken by a **staff member**.

The date and time of when the request is received (**time in**) and results are returned (**time out**) can be recorded.

Tests store different data depending on their type: an HIV test has different information from a Malaria test. Instead of statically defining all types of tests, a templating system was devised, where users can define what pieces of information can be stored for each type of test.

Tests are created in the **data entry** section of the system. Templates are created in the **administration** section

These test **templates** are identified by their **name**, and have an associated colour, which is used as the background **colour** when displaying the form for that test. Templates also have a **description** field.

Templates can have many **fields**, which can be put into **groups** (which aids data entry by splitting up fields visually).

Fields are identified by **name**, and can be flagged as a **required** field. Fields can be flagged as storing **numbers** or **text**. A **label** can be attached to a field, which is displayed after the form input element in data entry.

Input for a field can be made into a selectable list instead of a text box by defining a set of **limits** for the field. When limits exist, one can be made the **default** option. Additionally, fields can be flagged as accepting **multiple values** when limits exist.

Fields can have **subfields**. Subfields are displayed in data entry as being "children" of the main field. They can be displayed **inline** (next to) or **underneath** the parent field. Subfields can also be hidden from view, only being displayed when the parent value is within an **upper and lower limit**.

The level of subfields is infinite: subfields can have subfields of their own, and so on.

Templates can also have **subtests**. These are other templates that can be filled out at the same time, reducing the need to specify department, patient, staff and time data individually. Patients are usually tested for many things at once, and subtests helps make data entry quicker in this regard.

Reporting is an important part of any data collection system. Reports can be run on any type of test, providing a breakdown by patient age, gender and ethnicity. Department and staff breakdowns are also available. Finally, test-specific data (for example, the range of Haemoglobin values) are also available for the reporting set.

Patients are an important part of the system. Mae Tao Clinic has an electronic patient registry system managed by Health Information Services (HIS). The system retrieves patient data from this system using a Registration Number (**RN**), which all patients are given. When filling out a form in data entry, the user only needs to supply an RN, and all relevant patient data (name, age, gender, ethnicity, location) are automatically migrated. Furthermore, the system can retrieve a **patient's history** for all tests recorded in the system.

The danger with custom-built systems for data collection is "lock in": the raw data can be difficult to extract from the system in its entirety. This system provides a **data export** feature (in XML and Spreadsheet format). This means all information stored in the system can easily be extracted for a variety of uses: sharing with other organisations / individuals, or migrating to a new system in the future.

Technical information: software and hardware considerations

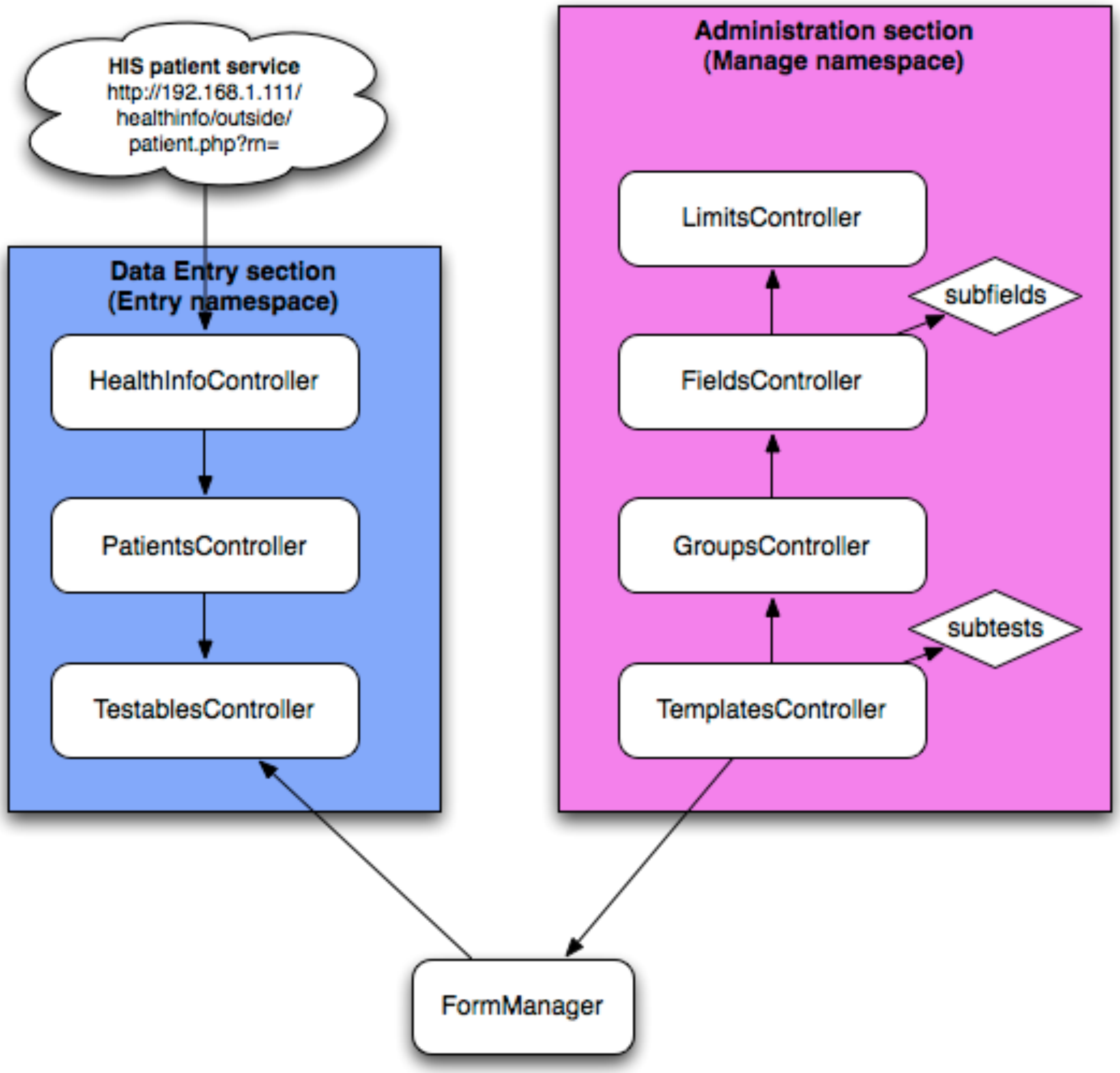
LATRS has been developed using **Ruby on Rails**, a software framework well suited to web applications. It is cross -platform, allowing for easy installation on Windows, UNIX and Mac environments. In a development mode, a simple database system (Sqlite) is used, while MySQL is used in production.

Git has been used to manage software control during the development of LATRS, and the entire codebase is available online via https://github.com/smcphill/LATRS. This allows software updates to be made available and deployed very easily.

For more information on the actual code, refer to the codebase documentation which can be generated using the **rdoc** utility.

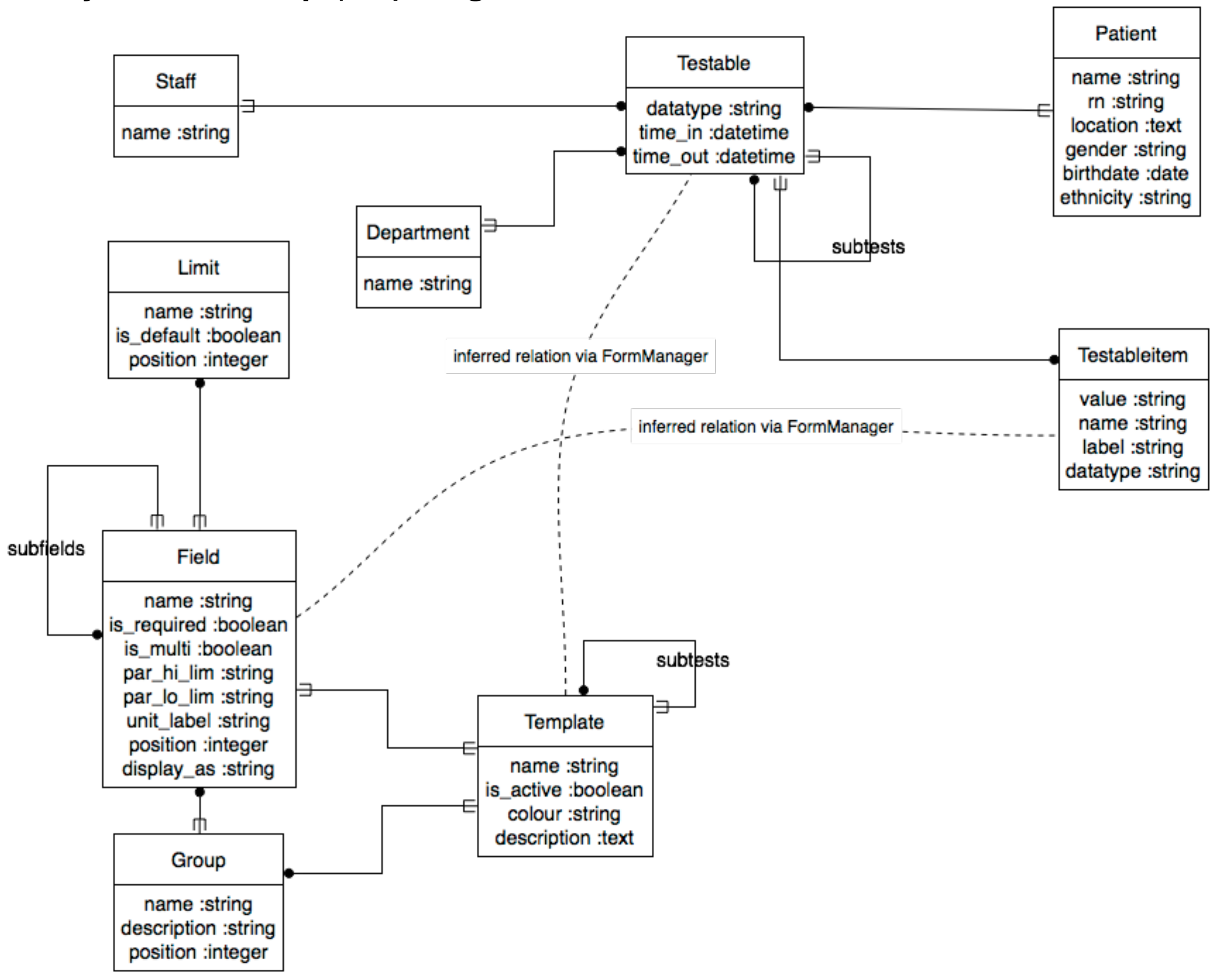
System diagrams

From templates to test results



Templates are created in the administration section. Active templates are registered with the FormManager, which in turn is used by the Data Entry section to know which types of tests can be created. The HealthInfoController class provides the link to the external patient registry, creating Patients as required.

Entity-Relationship (ER) diagram


Staff
name :string
Testable
datatype :string
time_in :datetime
time_out :datetime
Patient
name :string
rn :string
location :text
gender :string
birthdate :date
ethnicity :string
Department
name :string
subtests
Limit
name :string
is_default :boolean
position :integer
inferred relation via FormManager
Testableitem
value :string
name :string
label :string
datatype :string
inferred relation via FormManager
subfields
Field
name :string
is_required :boolean
is_multi :boolean
par_hi_lim :string
par_lo_lim :string
unit_label :string
position :integer
display_as :string
subtests
Template
name :string
is_active :boolean
colour :string
description :text
Group
name :string
description :string
position :integer