- Metis -

Localization Program Management System

Adam Lloyd, Web Development

Department of Computer Science, Cork Institute of Technology, Cork, Ireland

Intel Security (McAfee), City Gate, Mahon, Cork, Ireland

Introduction

For my final year project, I developed a program management system for the software localisation department at Intel Security (McAfee).

Their existing system was unsuitable for their current requirements:

- Workflow involved many manual steps .
- Duplication of data and extremely complex domain resulted in high error rate.
- Insecure code and outdated interface.

Developed for industry
Outdated system
Complex domain

Solution

The resulting system serves as a central repository where financial, project, and job information can be managed and accessed by Intel Security staff and applications (via API) .

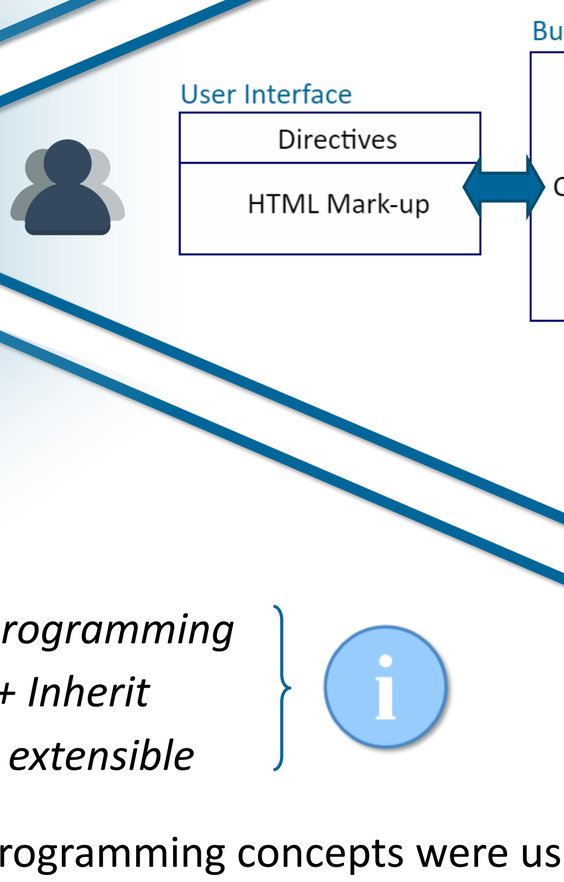
All stakeholders now reference the same information and their workflow is streamlined with a reduced error rate.

The modern, usable interface, secure and extensible codebase and high performance technologies it was developed with make for a robust system.

Improved workflow
Central repository
Modern Interface

Angular.js client providing a responsive interface.

API endpoints exposed by controllers implementing a generic class.



Angular Services make calls to API endpoints to pass JSON data to controllers.

Data access controlled by a generic repository to interact with the SQL data layer.

Generic programming
Abstract + Inherit
Concise + extensible

Agile methodology
Stakeholder meetings
Continuous integration

Several programming concepts were used to add consistency and security to the system by reducing code duplication.

- Generic programming enabled creation of thin controllers with shared common functionality from super classes and separation of controller specific functions.
- The data layer was abstracted using the repository pattern implementing a generic repository interface.
- Stateless token based authentication allows humans and machines (API) to access the system securely.

Architecture

The project was developed using the agile development methodology, scrum in particular, and integrated with Intel's continuous integration system.

- Components separated into 4 epics and managed in 3 week sprints using JIRA project management software.
- Weekly meetings with stakeholders.
- Jenkins build automation server monitored Git repository for changes and created builds for DEV and QA.

Development Process

Created With..

Acknowledgements

I would like to acknowledge the support of Michael McCourt, Angie O'Donnell, Rouslan Placella, Mary Davin, Ruairi O'Reilly and everyone at Intel/McAfee for their support.