Jacklyn Codebase

Build enterprise web applications with ease.

A quick overview

Version 1.0.1

January 2019

The Code Department

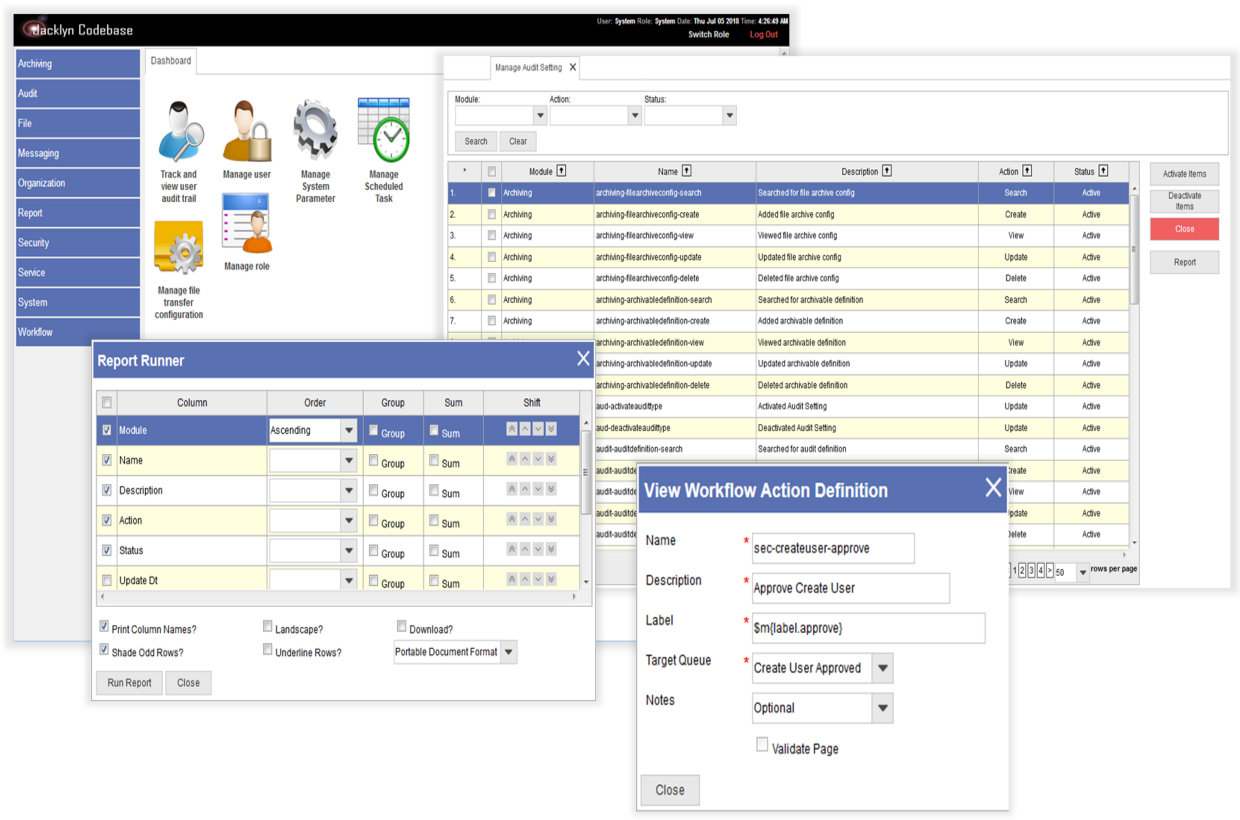
TCD

1.0 Introduction

Enterprise applications provide a modern platform on which organizations and businesses rely to perform their day-to-day operations. Usability, functional capacity and branch-wide availability of such applications determine how well these operations are executed and has a direct overall impact on the quality of goods and services provided by an enterprise to its customers.

As a business or software vendor with internal development teams, the technologies you adopt will influence the performance, robustness, reliability and scalability of the enterprise solutions you develop. They will also affect how your developers produce and maintain software. Adopting proven tools and technology that are easy to use and extend empowers your developers to build highly effective solutions as quickly as business requirements change.

Jacklyn Codebase is a component-based, customizable and extensible platform with pre-built functional modules that allows you to develop great production-ready enterprise web applications with ease.



Module: Action: Status:

*		Module	Name	Description	Action	Status
1.		Archiving	archiving-filearchiveconfig-search	Searched for file archive config	Search	Active
2.		Archiving	archiving-filearchiveconfig-create	Added file archive config	Create	Active
3.		Archiving	archiving-filearchiveconfig-view	Viewed file archive config	View	Active
4.		Archiving	archiving-filearchiveconfig-update	Updated file archive config	Update	Active
5.		Archiving	archiving-filearchiveconfig-delete	Deleted file archive config	Delete	Active
6.		Archiving	archiving-archivabledefinition-search	Searched for archivable definition	Search	Active
7.		Archiving	archiving-archivabledefinition-create	Added archivable definition	Create	Active
			archiving-archivabledefinition-view	Viewed archivable definition	View	Active
			archiving-archivabledefinition-update	Updated archivable definition	Update	Active
			archiving-archivabledefinition-delete	Deleted archivable definition	Delete	Active
			aud-activateaudittype	Activated Audit Setting	Update	Active
			aud-deactivateaudittype	Deactivated Audit Setting	Update	Active
			audit-auditdefinition-search	Searched for audit definition	Search	Active
			audit-auditde		reate	Active
			audit-auditde		View	Active
			audit-auditde		pdate	Active
			audit-auditde		elete	Active

1 2 3 4 > 50

Report Runner

	Column	Order	Group	Sum	Shift
✓	Module	Ascending	Group	Sum	
✓	Name		Group	Sum	
✓	Description		Group	Sum	
✓	Action		Group	Sum	
✓	Status		Group	Sum	
	Update Dt		Group	Sum	

[✓] Print Column Names? [ ] Landscape? [ ] Download?
[✓] Shade Odd Rows? [ ] Underline Rows? Portable Document Format

View Workflow Action Definition

Name *	sec-createuser-approve
Description *	Approve Create User
Label *	$m{label.approve}
Target Queue *	Create User Approved
Notes	Optional
	[ ] Validate Page

Close

2.0 Architecture

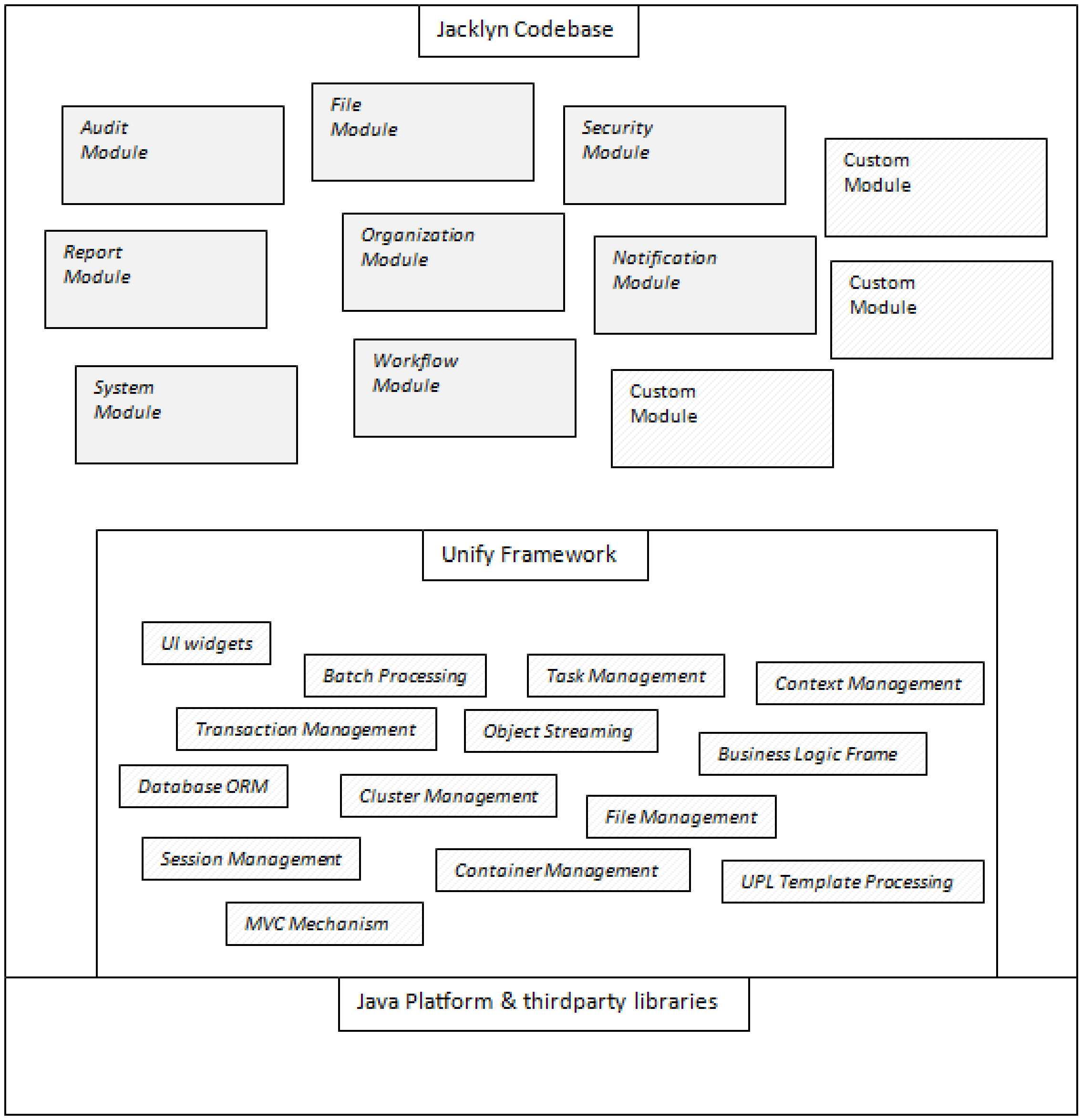


Fig. 2.1 Modular Architecture

3.0 Pre-built Functional Modules

3.1 Audit Module

This module that keeps track of and allows the examination or inspection of user related activities in the system.

Value: Applications manage critical information that an organization's personnel have access to and are required to manipulate as the perform day-to-day task. The capability of a global insight into all activities for user observation and accountability is a must have.

Highlights:

- Configurable global activity tracking
- Track user activity and behavior
- Inspect activities in chronological order performed on business entities

3.2 File Module

Used to manage system-wide file processing and transfer operations.

Value: Typical enterprise applications have to consume files (SWIFT and posting files for instance) received from external systems that contain information required for internal business processes. Also, in some cases, applications have to generate and transfer files to external systems.

Highlights:

- Define batch file structures
- Perform batch file read configuration
- Perform batch file uploads
- Configure file transfer operations
- Perform file transfer operation
- Setup automatic file processing or transfer operations

3.3 Notification Module

Used to manage message creation and dispatch within an application and to external systems.

Value: Provides a mechanism for internal system communication to application users and allows an organization to send formal notices to its clients as business operations progress; facilitating work and improving the overall quality of product and service delivery.

Highlights:

- Define notification message templates

- Configure notification channels (System, Email, SMS, etc)
- Send targeted or system wide notifications
- Setup automatic notifications

3.4 Organization Module

This module allows you to define an organizational structure in the system.

Value: Allows an application to mirror an organization's structure making it easier to define actors that perform different roles in an organization's business processes.

Highlights:

- Define and manage branch information
- Define and manage department information
- Define and manage role information including privileges

3.5 Report Module

Module for managing overall reporting capabilities of the system.

Value: Allows the extraction of useful information from your system in various formats.

Highlights:

- Configure reports including templates, input parameters and data sources
- Perform report generation
- Partial ad hoc report generation
- Multiple report formats (PDF, Excel, CSV, etc)
- Setup automatic report generation

3.6 Security Module

Used to manage user and client application access to system features and resources.

Value: Provides overall system access security.

Highlights:

- Create and manage user information
- Assign roles to users
- Create and manage remote client information
- Assign system resources and services to remote client
- Manage and control user sessions

3.7 System Module

This module is used to manage the configuration of variables and settings that determine system presentation, operation and behavior.

Value: A single point for system control.

Highlights:

- Maintain global application variables
- Manage application menu
- Manage application shortcuts
- Manage application dashboards
- Securely store system-wide authentication variables
- Setup scheduled tasks (Jobs)
- View scheduled task history
- Manage themes

3.8 Workflow Module

Used to manage the definition and execution of business processes that determine how a work item is processed from initiation to completion.

Value: Powerful workflow system that allows your application to easily mimic your client’s work items and business processes; with features that make it possible to adapt quickly as they change.

Highlights:

- Simple step-based workflow
- Workflow business rules
- Ad hoc document definition including document attachment
- Ad hoc form definition
- Step-specific fine-grained form section and widget control
- Notifications and alerts

4.0 Extending the Codebase

Jacklyn codebase, delivered as a set of jar files, is designed to be highly customizable and extensible, allowing you and your development team to easily create new applications by adding new features specific business requirements of your application. You can add new modules and extend any of the pre-built modules. You can also extend the workflow mechanism by define new workflow templates that capture business processes around work items.

Tools are available for code generation and workflow template modeling. You can generate entire Java source files, UPL templates and other ancillary files for a new module with CRUD functionality for selected database entities. You also have access to a visual workflow template designer.

4.1 Technology

Jacklyn codebase is built on a component framework that is based on the Java platform. Its software architecture adopts the popular 3-tier architecture for web applications which is composed of three layers of computing – the presentation tier, business logic tier and data tier. The MVC model is used in the presentation tier with view elements based on UPL templates.

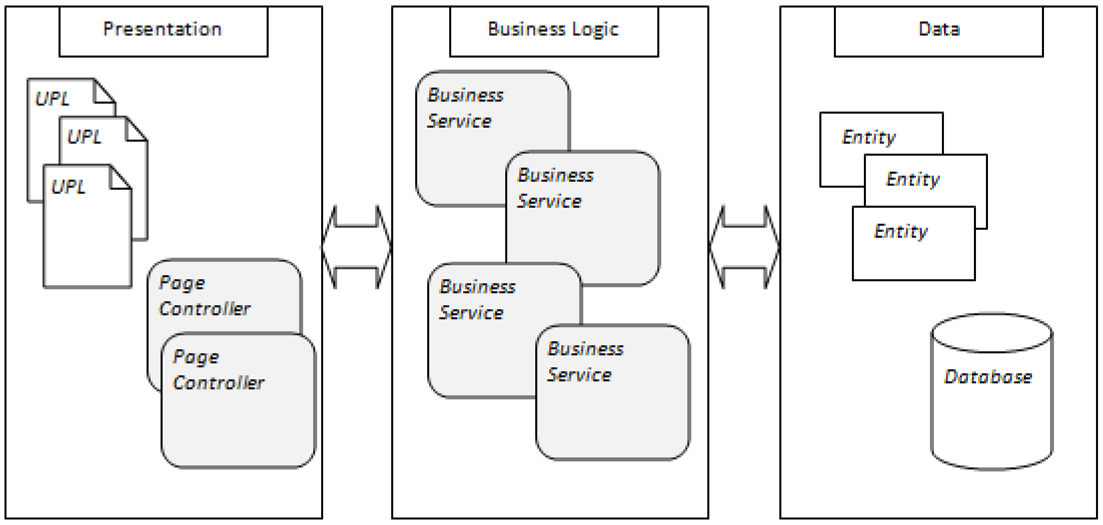


Fig. 4.1 Software architecture

4.1.1 Sample

Below are artifact listings used to add CRUD functionality for a Book entity in an illustrative inventory system.

Listing 4.1 Book.java

```
@Table
public class Book extends BaseEntity {

    @Column
    private String title;

    @Column
    private String author;

    @Column
    private BigDecimal price;
```

```
@Column
private Integer copiesInStock;

@Column
private Date salesExpirationDt;

public String getDescription() {
    return title;
}

public String getTitle() {
    return title;
}

public void setTitle(String title) {
    this.title = title;
}

public String getAuthor() {
    return author;
}

public void setAuthor(String author) {
    this.author = author;
}

public BigDecimal getPrice() {
    return price;
}

public void setPrice(BigDecimal price) {
    this.price = price;
}
```

```java
    public Integer getCopiesInStock() {
        return copiesInStock;
    }

    public void setCopiesInStock(Integer copiesInStock) {
        this.copiesInStock = copiesInStock;
    }

    public Date getSalesExpirationDt() {
        return salesExpirationDt;
    }

    public void setSalesExpirationDt(Date salesExpirationDt) {
        this.salesExpirationDt = salesExpirationDt;
    }

}
```

Listing 4.2 BookQuery.java

```java
public class BookQuery extends BaseEntityQuery<Book> {

    public BookQuery() {
        super(Book.class);
    }

    public BookQuery titleLike(String title) {
        return (BookQuery) like("title", title);
    }

    public BookQuery authorLike(String author) {
        return (BookQuery) like("author", author);
    }
}
```

Listing 4.3 InventoryService.java

```java
public interface InventoryService extends JacklynBusinessService {

    Long createBook(Book book) throws UnifyException;

    Book findBook(Long bookId) throws UnifyException;

    List<Book> findBooks(BookQuery query) throws UnifyException;

    int updateBook(Book book) throws UnifyException;

    int deleteBook(Long bookId) throws UnifyException;
}
```

Listing 4.4 InventoryServiceImpl.java

```java
@Transactional
@Component("InventoryService")
public class InventoryServiceImpl extends
      AbstractJacklynBusinessService implements InventoryService {

    public Long createBook(Book book) throws UnifyException {
        return (Long) db().create(book);
    }

    public Book findBook(Long bookId) throws UnifyException {
        return db().find(Book.class, bookId);
    }

    public List<Book> findBooks(BookQuery query)
      throws UnifyException {
        return db().findAll(query);
    }
    public int updateBook(Book book) throws UnifyException {
        return db().updateById(book);
    }

    public int deleteBook(Long bookId) throws UnifyException {
        return db().delete(Book.class, bookId);
    }
}
```

Listing 4.5 BookController.java

```
@Component("/inventory/book")
@UplBinding("web/inventory/upl/managebook.upl")
public class BookController extends BaseCrudController<Book, Long> {

    @Configurable
    private InventoryService inventoryService;

    private String searchTitle;

    private String searchAuthor;

    public BookController() {
        super(Book.class, "book.hint", ManageRecordModifier.CRUD);
    }

    public String getSearchTitle() {
        return searchTitle;
    }

    public void setSearchTitle(String searchTitle) {
        this.searchTitle = searchTitle;
    }

    public String getSearchAuthor() {
        return searchAuthor;
    }

    public void setSearchAuthor(String searchAuthor) {
        this.searchAuthor = searchAuthor;
    }

    @Override
    protected List<Book> find() throws UnifyException {
        BookQuery query = new BookQuery();
        if (searchTitle != null) {
            query.titleLike(searchTitle);
        }

        if (searchAuthor != null) {
            query.authorLike(searchAuthor);
        }
        query.ignoreEmptyCriteria(true);
        return inventoryService.findBooks(query);
    }

    @Override
    protected Book find(Long id) throws UnifyException {
        return inventoryService.findBook(id);
    }

    @Override
    protected Book prepareCreate() throws UnifyException {
        return new Book();
    }

    @Override
    protected Object create(Book book) throws UnifyException {
        return inventoryService.createBook(book);
    }
    @Override
    protected int update(Book book) throws UnifyException {
        return inventoryService.updateBook(book);
    }

    @Override
    protected int delete(Book book) throws UnifyException {
        return inventoryService.deleteBook(book.getId());
    }

}
```

Listing 4.6 managebook.upl

```
//Manage books page UPL template
!ui-page caption:$s{Manage Books}
    searchByList:$c{searchTitle searchAuthor}
    searchClearList:$c{searchTitle searchAuthor}
    tableColumnList:$c{titleCol authorCol copiesInStockCol}
    formColumns:1
    formSection:$d{!ui-section caption:$s{Basic Details}
        components:$c{frmTitle frmAuthor frmPrice frmCopiesInStock
           frmSalesExpirationDt}}

//Search
!ui-text:searchTitle caption:$s{Title} binding:searchTitle
    eventHandler:$d{!ui-event event:onenter action:$c{searchAct}}
!ui-text:searchAuthor caption:$s{Author} binding:searchAuthor
    eventHandler:$d{!ui-event event:onenter action:$c{searchAct}}

//Table columns
!ui-label:titleCol caption:$s{Title} binding:title
    columnStyle:$s{width:250px;} sortable:true
!ui-label:authorCol caption:$s{Author} binding:author
```

```
    columnStyle:$s{width:250px;} sortable:true
!ui-label:copiesInStockCol caption:$s{No. of Copies}
     binding:copiesInStock
     style:$s{text-align:right;} columnStyle:$s{width:100px;}
     sortable:true


//Form components
!ui-text:frmTitle caption:$s{Title} binding:title required:true
     focus:true
!ui-text:frmAuthor caption:$s{Author} binding:author required:true
!ui-decimal:frmPrice caption:$s{Price} binding:price precision:6
     scale:2 required:true
!ui-integer:frmCopiesInStock caption:$s{No. of Copies}
     binding:copiesInStock precision:4 required:true
!ui-date:frmSalesExpirationDt caption:$s{Expiration Date}
     binding:salesExpirationDt required:true
```

5.0 Getting started

Take the codebase for a spin!

Download the latest Jacklyn starter application build from https://github.com/tcdng/jacklyn-app/releases/download/1.0.1/jacklyn-app-starter-1.0.1.zip .

Get brief document on steps to run here https://github.com/tcdng/jacklyn-app/files/2777020/Run.Jacklyn.Starter.1.0.1.pdf .

View framework and platform repositories here https://github.com/tcdng .

6.0 Licensing

Jacklyn Codebase is released under Apache 2.0 open source license which means you can use it worry-free in private and commercial projects.

7.0 About

At The Code Department, we are passionate about software, its development and how it is applied to providing superb, high quality and practical solutions to enterprises in various industries.

We develop technologies and tools that make this happen; around which we offer commercial support and software development services.