



APPLICATION FOR REMOTE PLANE GEOMETRY TEACHING

Vũ Việt Tùng

Trần Văn Quang

Nguyễn Thế Minh Nhật

Lưu Võ Ngọc Châu

Supervised by **Thân Văn Sử**

No. SE04.A - 002



MINISTRY OF EDUCATION AND
TRAINING

FPT UNIVERSITY

Capstone Project Document

Application For Remote Plane Geometry Teaching

SE04.A – 002	
Group Members	Vũ Việt Tùng – 60244 Trần Văn Quang – 60077 Nguyễn Thế Minh Nhật – 60067 Lưu Võ Ngọc Châu – 60076
Supervisor	Thân Văn Sử
Capstone Project code	RPG

- Ho Chi Minh City, April/2012 -

Contents

Introduction

Software Project Management Plan

Problem Definition.....	2
Name of this Capstone Project.....	2
Project Overview.....	2
Project organization.....	3
Software Process Model	3
Role and Responsibilities	3
Tools and Techniques.....	3
Project Management Plan.....	4
Coding Conventions.....	6
Software Requirements Specifications	7
User Requirement Specification.....	7

System Requirement Specification

External Interface Requirement	8
System Features.....	8
System features – Account.....	10
System features – Chat.....	13
System features – Room Management.....	15
System features – Sheet Management.....	19
System features – Contact Management	24
System features – Sharing.....	26
System features – Drawing	31
Software System Attributes.....	34
Entity Relationship Diagram.....	35

Software Design Description

Design Overview	36
System Architectural Design.....	36
Component Diagram.....	38
Detail Description of Components.....	39
Common Library.....	39

Silverlight Library	60
Pythagore Service.....	71
Silverlight Client.....	76
Sequence Diagrams	77
User Interface Design	83
Database Design.....	88

Software Test Documentation

Introduction.....	90
Test Plan	90
Test Cases.....	91
Account Features.....	91
Chat Features	93
Room management Features.....	94
Sheet management Features.....	97
Contact management Feature.....	99
Share Features	101
Draw Features	103
Checklists.....	111
Feature account.....	111
Feature chat	113
Room management.....	113
Sheet management.....	115
Contact management	116
Share feature.....	117
Draw feature.....	118

Software User's Manual

Installation Guide.....	133
User's Guide	136

Appendix

Supported Drawing tools.....	A
Task sheets.....	C

Introduction

In Vietnam, online tutoring has become a popular trend for foreign language study. But its counterparts in mathematics and other scientific subjects is not so widespread. One of the main reasons is the lack of tools to prepare and deliver lessons over the network. Additionally, the lack of digital study resources is the obstacle for mathematical remote teaching from further achievements.

From these initial ideas, we decided to build an application to resolve the aforementioned problems, which can bring Vietnam education to a new level. The principles of this project are creating a new tool for high school mathematics teachers to prepare and deliver the geometry lessons to students with ease.

Software Project Management Plan

Problem Definition

Name of this Capstone Project

Project Full name: **Application for remote plane geometry teaching.**

Project code: **RPG** (Remote Plane Geometry)

Software name: **Pythagore**

NOTE:

From this line,
These 3
names can be
used
interchangeab-
ly.

Project Overview

The Current System

The most frequently used manner to create geometry lessons is to use Microsoft™ Paint® to draw line and text, which has many disadvantages, enlisted as follows:

1. Segment lengths and angle magnitudes are not accurate.
2. Inflexibility for changing.
3. Once saved, everything becomes an image.
4. No collaboration.

More flexible drawing applications (ex: Adobe Photoshop) can be used. However, it will reduce the usability.

The Proposed System

New system will have the ability to:

1. Draw objects that have accuracy in length and magnitude.
2. Objects should be easy to be changed after being created or saved.
3. All drawings are saved on a remote server and can be downloaded in specified format.
4. Provide a manner for multiple users to view and, if possible, can make their own changes.
5. Saved drawings can be easily shared publicly or to private groups.

Boundaries of the System

1. Only used in Geometry for high schools.
2. Only focus on building a tool and a framework for preparing and delivering lessons.
3. User interface language is in Vietnamese (some technical terms are in English).

Project organization

Software Process Model

Due to report submission schedule, this project used waterfall as its process model.
Waterfall model includes:

Requirement Analysis and Definition: All possible requirements of the system to be developed are captured in this phase.	System and Software Design: The requirement specifications are studied to understand the requirements and have an idea of how should the end product looks like.
Implementation and Unit Testing: The system is first developed in small programs called units. Each unit is developed and tested for its functionality.	Integration and System Testing: Units are integrated into a complete system during integration phase and tested to check if all modules/units coordinate with each other.

Role and Responsibilities

Table 1 - Roles and Responsibilities

Name	Roles	Responsibilities
Thân Văn Sứ	Project manager	Assistant Director of Identity Management
Vũ Việt Tùng	Team leader	Process control and audit, support on process related issues
Nguyễn Thế Minh Nhật	Team member	Engineers, Training, Customer Support
Lưu Võ Ngọc Châu	Team member	Engineers, Training, Customer Support
Trần Văn Quang	Team member	Engineers, Training, Customer Support

Tools and Techniques

Software

1. Microsoft Visual Studio 2010: Use to code software modules
2. Microsoft SQL server 2008: Use to prepare the database for the application
3. Microsoft Project : For the team leader to manage the work of the members and the process of project
4. TortoiseSVN: Control Source code of the whole project
5. VisualSVN: extension for using subversion (SVN) inside Visual Studio
6. Google Cloud Connect: Connect and synchronize the documents

Hardware

Laptop or PC with appropriate environment for developing and testing

Project Management Plan

Tasks

Table 2 - Tasks

Feasibility study	
Description	Market and theory research, technology & business process study
Deliverables	The feasibility report and decisions for the project
Resources Needed	13 man-day
Dependencies and Constraints	N/A
Risks	The project or the chosen technology is not feasible.
Documentation and review	
Description	Create all the necessary documents for research and delivery
Deliverables	<ol style="list-style-type: none"> 1. Project Management Plan (PMP) 2. Software Requirements Specification (SRS) 3. Software Design Description (SDD) 4. Software Test Documentation (STD) 5. Software User's Manual (SUM)
Resources Needed	FPT templates, 48 man-day
Dependencies and Constraints	Follow FPT templates
Risks	Not follow FPT templates, poor review leads to faults in later phases
GUI design and implementation	
Description	Design user interface
Deliverables	Prototype
Resources Needed	7 man-day
Dependencies and Constraints	Metro style
Risks	The interface is not usable& ineffective
Implementation	
Description	Create the executable files
Deliverables	Installation guide for web server
Resources Needed	Visual Studio 2010, Blend 4, Silverlight 4, .NET framework 4 150 man-day
Dependencies and Constraints	N/A
Risks	Not done before deadline, or not all the functions can be implemented
Integration	
Description	Integrate all the modules
Deliverables	The whole Pythagore package
Resources Needed	All the modules 10 man-day

Dependencies and Constraints	N/A
Risks	The modules do not work together
Release and deployment	
Description	Release the complete application and deploy it for the customer
Deliverables	Pythagore installation file and all the related documents (SUM)
Resources Needed	Installation package 5 man-day
Dependencies and Constraints	Meet the user requirements
Risks	The program is not running properly in real environment Cannot release before the deadline
Quality control	
Description	Testing application's performance and usability
Deliverables	STD
Resources Needed	QA, testers (team member),members, FPT template test case 146 man-day
Dependencies and Constraints	Follow FPT template test case
Risks	The program performance bellow standard Not user friendly Bugs and leakages
Human resource management	
Description	Manage human resource, task assignments and member's performances
Deliverables	Project Task List – Assignment Table sheet
Resources Needed	Project Task List
Dependencies and Constraints	N/A
Risks	Lack of planning and management skill Communication problem Decrease in team members during project implementation

Task sheet: Assignment and Timetable

See Table 10 (page C)

Coding Conventions

Naming conventions

- Private Fields: underscore followed by lowerCamelCase.
- Non-private Fields and properties: UpperCamelCase.
- Local variables: lowerCamelCase.
- Do not use consecutive underscores in name.
- Do not use Hungarian style.

Lengths

- Public name should not be longer than 32 characters or 7 words.
- Methods should contain no more than 70 lines of code (if it is, it must be divided into methods).
- Methods should contain no more than 5 levels of indentation (if it is, it must be divided into methods).
- A line of code should contain no more than 80 characters.

Other convention

- Use **var** keyword whenever possible.
- Remove all unnecessary **using** in file header.
- Do not allow empty **catch** block.

Software Requirements Specifications

User Requirement Specification

Table 3 - User's Possibilities

User's Possibilities	
Drawing	<ul style="list-style-type: none">- Draw geometry objects quickly and accurately.- Drawn objects can be moved and have its properties (thickness, color, name, etc.) changed easily.- User can easily undo and redo his/her actions.- User can capture the current sheet into an image file or download it as a XML file- At one time, a user can draw many figures on one or many sheets
Sharing and Collaborating	<ul style="list-style-type: none">- User can have other people to see what he/she is doing- Other participants can contribute to the drawing- All participants can chat- User who created a sheet can embed that sheet into a webpage as-is.- Application should also allow creator of a sheet to broadcast that sheet on any webpage so viewer can see what is changing.- User can manage their previous sharing actions.
Account managing	<ul style="list-style-type: none">- User can save what he/she did for later revision and modification- User can change his/her profile information- User can add other users as friend so they can easily invite their friends to sheets- Friends can be grouped

System Requirement Specification

External Interface Requirement

User Interfaces

The user can interact with the system and other users through the user interface. There are different screens available for the users to enter the details. Error messages are also generated. Below are screens available for users:

Table 4 - Screens

Screen name	Function
Login screen	Allow registered users to login to the system
Register screen	Allow guests to register a username
Edit information screen	Allow logged users to change basic information, includes display names, email, etc.
Home screen	Allow logged users to view and manage their contacts, rooms, shared sheets.
Room screen	Allow logged user to draw, view what other participants draw, and send messages
Embedded Screen	Allow everyone to see and interact (if allowed) with shared sheets

NOTE:
See User Interface Design section (page 83) for detail design and layout for screens

Hardware Interfaces

Network cable, network interface card

Software Interfaces

It will run comfortably on a web browser with a Silverlight plugin

Communication protocols

The Protocol to be used is TCP/IP with SOAP

System Features

This section includes all features (group of similar use cases) that this software will provide to satisfy user requirements. There are 5 kinds of user (actor), who will interact directly with the system:

Table 5 - Actors

Actor	Description
Guest	Visitor that is not logged in the system
User	Logged-in guest
Participant	User that currently participating in a room
Creator	The participant that creates a room or shares a sheet. Creator of a room is the creator of its sheets
Broadcaster	A participant that currently has the permission to draw and broadcast his/her drawing to other participants

Systems features and use cases are depicted in the following diagram:

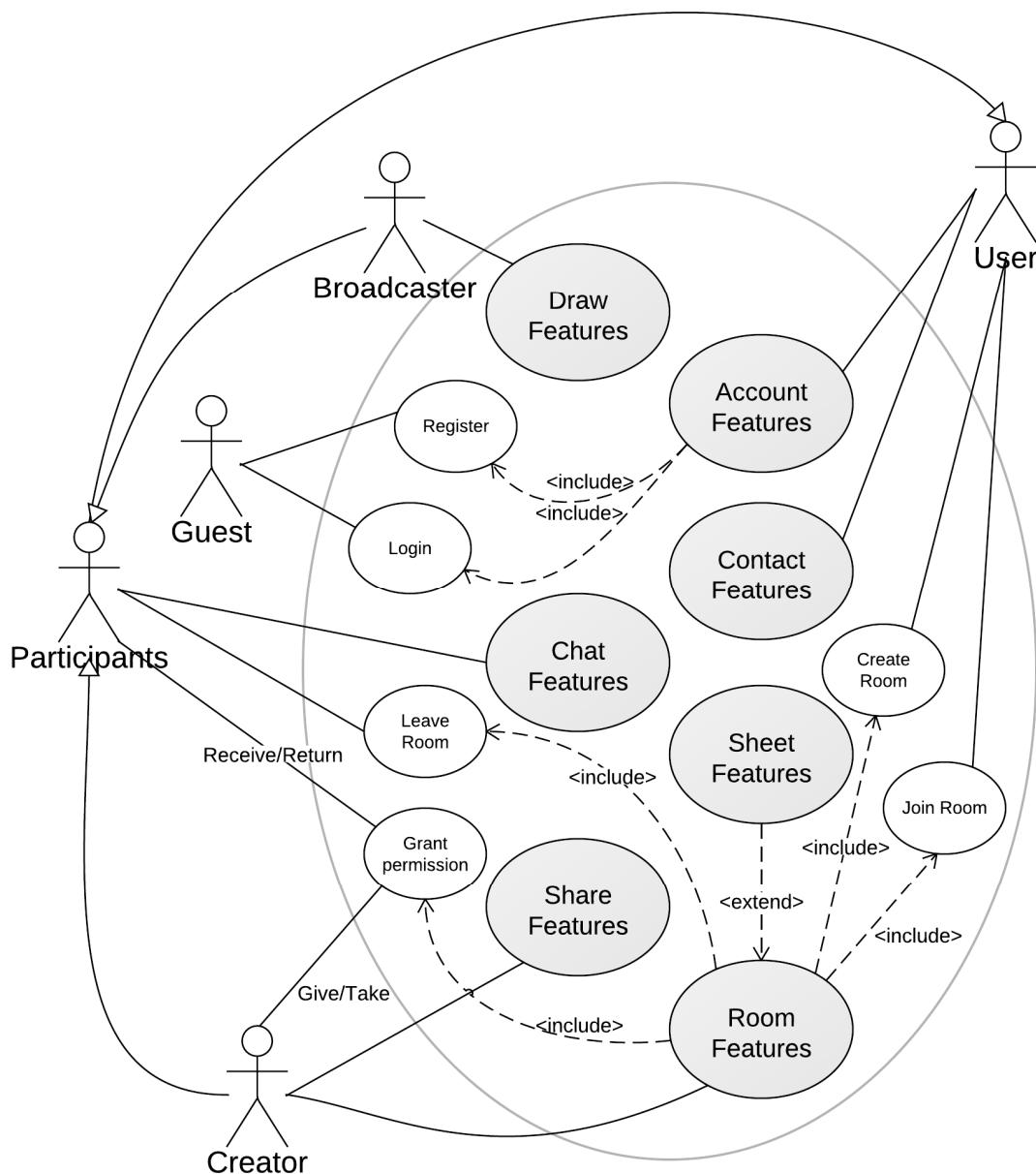


Figure 1 – Overall Use-Case diagram

System features – Account

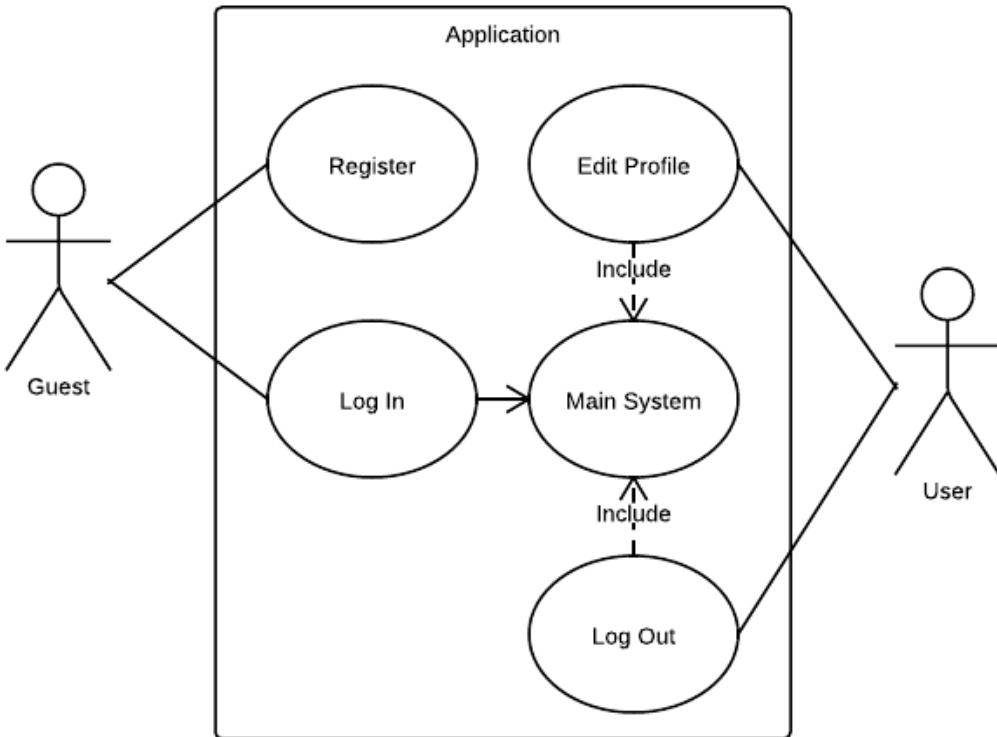


Figure 2 – Account feature use-case diagram

USE CASE-REGISTER SPECIFICATION			
Use-case No.	UC001	Use-case Version	1.0
Use-case Name	Account Register		
Author	NhatNTM		
Date	30/01/2012	Priority	Normal
Actor:	Guest		
Summary:	This use case is about account registering.		
Goal:	Guest can sign up an account into system database to log in later.		
Triggers:	Actor's behavior		
Pre-conditions:	N/A		
Post-conditions:	An account is registered successfully		
Main Success Scenario:	<ol style="list-style-type: none"> 1. Guest clicks Register link from Login Screen 2. Guest enters required information and confirms password. 3. Guest clicks Register button. 4. Guest is navigated to Login Screen 		

Alternative Scenario:N/A

Exceptions: When username and/or email are not available, guest will get notified and be required to enter valid information

Relationships: N/A

Business Rules: Username must not be longer than 32 characters, only A-Z, a-z, numeric and underscore is allowed. Underscore is not at the start, the end and not following each other. Password must be between 8-32 characters.

USE CASE-LOG IN SPECIFICATION

Use-case No.	UC002	Use-case Version	1.0
Use-case Name	Account Log In		
Author	NhatNTM		
Date	31/01/2012	Priority	Normal

Actor: Guest

Summary: This use case is about account logging in.

Goal: Allow system to be secured against unauthorized user.Guest can log in to the system to access other system features.

Triggers: Actor's behavior

Pre-conditions: N/A

Post-conditions: Guest is logged in

Main Success Scenario:

1. Guest enters username and password to respective input
2. Guest hits enter or click Login button

Alternative Scenario: N/A

Exceptions: When username and password are not available or not matched, guest will get notified and be required to enter valid information

Relationships: Require for all other features, except account register feature.

Business Rules: N/A

USE CASE-LOG OUT SPECIFICATION

Use-case No.	UC003	Use-case Version	1.0
Use-case Name	Account Log Out		
Author	NhatNTM		
Date	29/01/2012	Priority	Normal
Actor:	User		

Summary:	This use case is about signing out system.
Goal:	User can quit the system without leaving trace of account information.
Triggers:	Actor's behavior
Pre-conditions:	N/A
Post-conditions:	User status is inactive.
Main Success Scenario:	<ol style="list-style-type: none"> 1. User clicks logout button from Home Screen 2. User will be navigated to Login Screen
Alternative Scenario:	N/A
Exceptions:	N/A
Relationships:	Log Out feature is included in Log In feature.
Business Rules:	Logging out will invalidate sessions from all other locations.

USE CASE-EDIT PROFILE SPECIFICATION			
Use-case No.	UC004	Use-case Version	1.0
Use-case Name	Account Edit Profile		
Author	NhatNTM		
Date	03/02/2012	Priority	Normal
Actor:	User		
Summary:	This use case is about editing account information.		
Goal:	A mean for user to edit his/her own account information.		
Triggers:	Actor's behavior		
Pre-conditions:	N/A		
Post-conditions:	Account is edited.		
Main Success Scenario:	<ol style="list-style-type: none"> 1. User clicks edit information link and will be navigated to Edit information screen 2. User enters new information and click Save 3. User gets notified that operation is completed 		
Alternative Scenario:	N/A		
Exceptions:	N/A		
Relationships:	Account Edit feature is included in Log In feature.		
Business Rules:	N/A		

System features – Chat

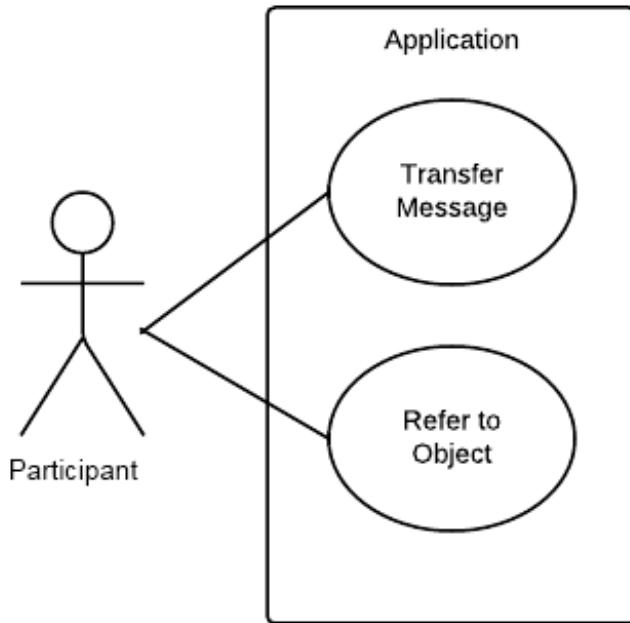


Figure 3 – Chat feature use-case diagram

USE CASE-CHAT MESSAGE TRANSFER SPECIFICATION			
Use-case No.	UC005	Use-case Version	1.0
Use-case Name	Message Transfer		
Author	NhatNTM		
Date	31/01/2012	Priority	Normal
Actor:	Participant		
Summary:	This use-case is about transferring message through chat.		
Goal:	Participant can send message, the message is then sent to everyone in the room.		
Triggers:	Actor's behavior		
Pre-conditions:	N/A		
Post-conditions:	All messages must be delivered to the right users		
Main Success Scenario:			
1.	Participant A: type a message to message box, then click send		
2.	All other participants in this room: Receive that message.		
Alternative Scenario:	N/A		
Exceptions:	N/A		
Relationships:	N/A		
Business Rules:	N/A		

USE CASE-CHAT REFER OBJECT SPECIFICATION			
Use-case No.	UC006	Use-case Version	1.0
Use-case Name	Refer Object		
Author	NhatNTM		
Date	31/01/2012	Priority	Normal
Actor:	Participant		
Summary:	This use-case is about referring to draw instance, which appear as hyperlinked text in chat box.		
Goal:	User can see where the hyperlinked text refers to which figure instance in draw screen.		
Triggers:	Actor's behavior		
Pre-conditions:	N/A		
Post-conditions:	N/A		
Main Success Scenario:	<ul style="list-style-type: none"> - When user types "@" in chat textbox, a suggestion box appears - User can choose an object from the list or keep typing to filter this list - Selected object appears in the textbox as a link. - User clicks on the link to highlight the object 		
Alternative Scenario:	<ul style="list-style-type: none"> - If the referred object is deleted, it will appear as normal text 		
Exceptions:	N/A		
Relationships:	N/A		
Business Rules:	N/A		

System features – Room Management

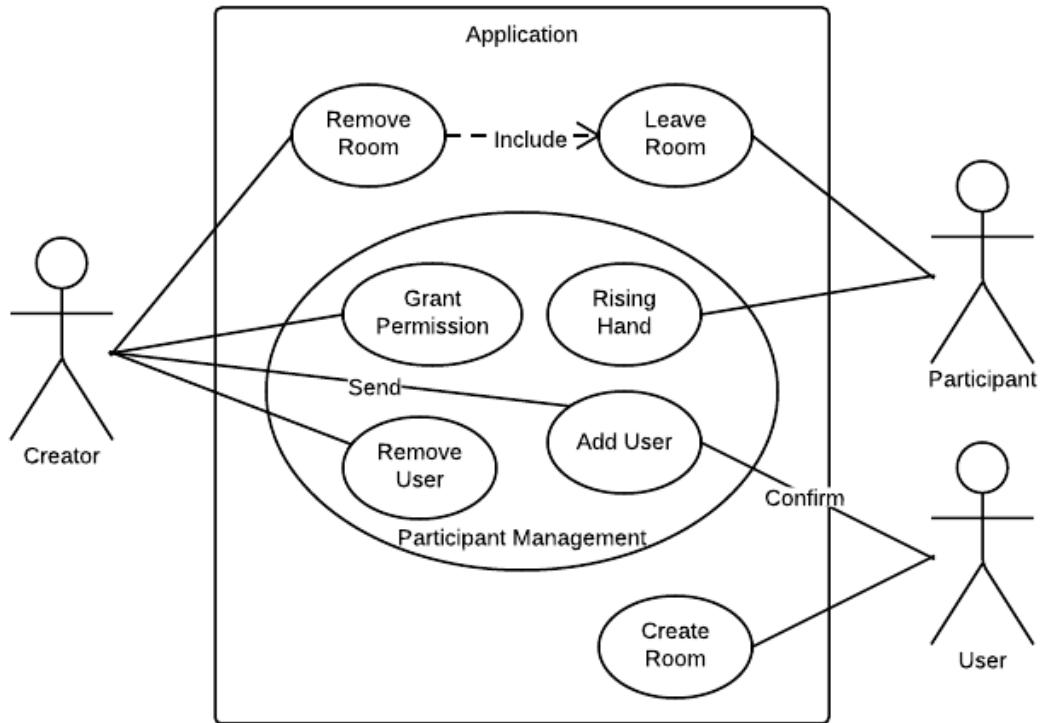


Figure 4 – Room management feature use-case diagram

USE CASE-CREATE ROOM SPECIFICATION			
Use-case No.	UC007	Use-case Version	1.0
Use-case Name	Create Room		
Author	NhatNTM		
Date	31/01/2012	Priority	Normal
Actor:	User		
Summary:	This use case is about how creator creates new room.		
Goal:	Create can create new room, allow other user to participate, required by drawing, chatting, embedding.		
Triggers:	Actor's behavior		
Pre-conditions:	N/A		
Post-conditions:	Room is created.		
Main Success Scenario:	<ol style="list-style-type: none"> 1. User clicks Create room button 2. User is prompted to enter room name 3. User clicks OK 4. New room appears in user room list 		

Alternative Scenario:	N/A
Exceptions:	N/A
Relationships:	Required by Draw, Chat, Sheet, Share.
Business Rules:	N/A.

USE CASE-REMOVE ROOM SPECIFICATION

Use-case No.	UC008	Use-case Version	1.0
Use-case Name	Remove Room		
Author	NhatNTM		
Date	31/01/2012	Priority	Normal

Actor: Creator

Summary: This use case is about how creator removes rooms.

Goal: Close the created room. While doing so, force creator and other remaining participants who still remain to leave the room.

Triggers: Actor's behavior

Pre-conditions: N/A

Post-conditions: Room is successfully removed

Main Success Scenario:

1. Creator clicks delete button next to room name in room list
2. A confirmation box appears
3. Creator clicks OK
4. Room is removed from room list, everyone in room gets notified that the room is successfully removed.

Alternative Scenario: N/A

Relationships: N/A

Business Rules: N/A.

USE CASE-ADD USER SPECIFICATION

Use-case No.	UC009	Use-case Version	1.0
Use-case Name	Add User		
Author	NhatNTM		
Date	31/01/2012	Priority	Normal
Actor:	Creator, User		

Summary:	This use case is about how creator invites other users to join.
Goal:	Other users can join into creator's room after confirming.
Triggers:	Actor's behavior
Pre-conditions:	N/A
Post-conditions:	User who agree to the invitation will join, do nothing otherwise
Main Success Scenario:	<ol style="list-style-type: none"> 1. Creator enters desired participant's username into a textbox 2. Creator clicks Invite 3. Invited participant gets a notification
Alternative Scenario:	N/A
Exceptions:	N/A
Relationships:	N/A
Business Rules:	N/A

USE CASE-REMOVE USER SPECIFICATION			
Use-case No.	UC010	Use-case Version	1.0
Use-case Name	Remove User		
Author	NhatNTM		
Date	31/01/2012	Priority	Normal
Actor:	Creator		
Summary:	This use case is about how creator remove participant out of room.		
Goal:	Creator can remove participant, force him/her out of room.		
Triggers:	Actor's behavior		
Pre-conditions:	N/A		
Post-conditions:	The chosen Participant must be successfully removed from room		
Main Success Scenario:	<ol style="list-style-type: none"> 1. Creator clicks on the x button next to participant name 2. Removed participant gets notification and force to leave the room 3. Everyone in room gets notification that participant has left room 		
Alternative Scenario:	N/A		
Exceptions:	N/A		
Relationships:	N/A		
Business Rules:	N/A		

USE CASE-LEAVE ROOM SPECIFICATION			
Use-case No.	UC011	Use-case Version	1.0
Use-case Name	Leave Room		
Author	NhatNTM		
Date	31/01/2012	Priority	Normal
Actor:	Participant		
Summary:	This use case is about how user leaves the room.		
Goal:	User is no longer inside the room they want to leave.		
Triggers:	Actor's behavior Triggered by remove user use case. Triggered by remove room use case.		
Pre-conditions:	N/A		
Post-conditions:	N/A		
Main Success Scenario:	<ol style="list-style-type: none"> User clicks leave button next to room name in room list in Home Screen User gets notified that he/she has left the room successfully Everyone in room gets notified that that user has left the room. 		
Alternative Scenario:	N/A		
Exceptions:	N/A		
Relationships:	N/A		
Business Rules:	Left user can no longer enter a room until a new invitation from creator		

USE CASE-GRANT DRAW PERMISSION SPECIFICATION			
Use-case No.	UC012	Use-case Version	1.0
Use-case Name	Grant Draw Permission		
Author	NhatNTM		
Date	02/02/2012	Priority	Normal
Actor:	Creator		
Summary:	This use case is about Creator allowing or disallowing a participant in a room to draw. This applies to all sheets within that room.		
Goal:	User is given or creator takes permission to draw.		
Triggers:	Actor's behavior		
Pre-conditions:	Only room creator has the permission to give or take permission		

Post-conditions:	After giving permission to one Participant, this Participant becomes the only broadcaster.
Main Success Scenario:	
1. Creator clicks on "pencil" button next to participant's name. 2. Current broadcaster is forced to stop drawing. 3. Everyone in room gets notification about the change	
Alternative Scenario: N/A	
Exceptions: User is no longer in room.	
Relationships: N/A	
Business Rules: Creator cannot disallow his/herself.	

USE CASE-RISING HAND SPECIFICATION						
Use-case No.	UC013	Use-case Version	1.0			
Use-case Name	Rising Hand					
Author	NhatNTM					
Date	02/02/2012	Priority	Normal			
Actor:	Participant					
Summary:	This use case is about participant want to get everyone in room's attention.					
Goal:	User participant giving signal to creator screen, thus putting a hand icon beside user name is participant list.					
Triggers:	Actor's behavior					
Pre-conditions:	User is logged in.					
Post-conditions:	N/A					
Main Success Scenario:						
Participant clicks on Rising Hand button, other participants can see an indication.						
Alternative Scenario: N/A						
Exceptions: Creator cannot raise hand						
Relationships: N/A						
Business Rules: Interval between rising hand attempts is 10 seconds.						

System features – Sheet Management

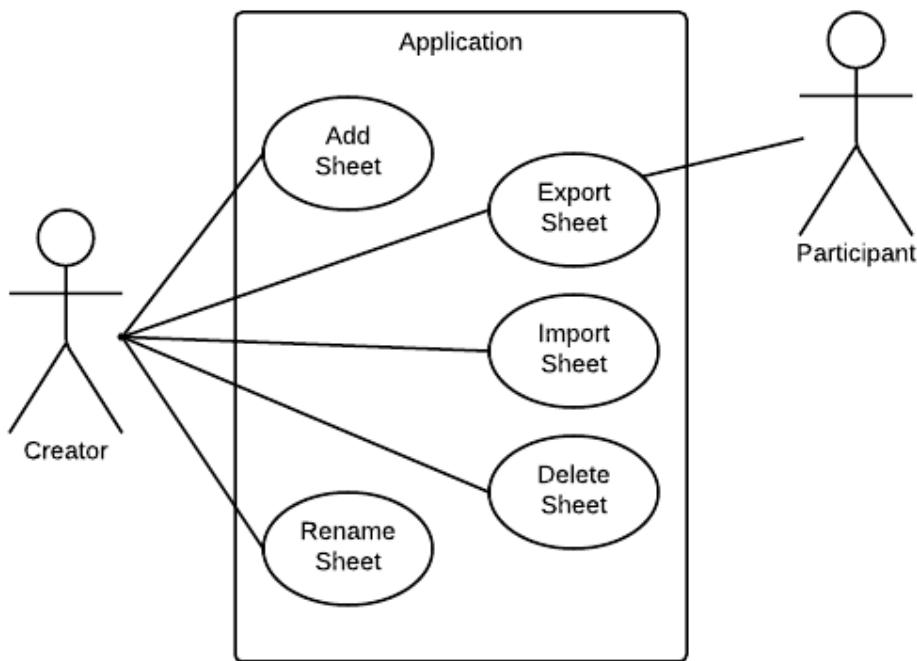


Figure 5-Sheet Management Feature Use-case diagram

USE CASE-ADD SHEET SPECIFICATION			
Use-case No.	UC014	Use-case Version	1.0
Use-case Name	Add Sheet		
Author	NhatNTM		
Date	02/01/2012	Priority	Normal
Actor:	Creator		
Summary:	This use case is about adding new sheet in a room.		
Goal:	Creator can create new sheet at the end, allow them to use draw and embed function.		
Triggers:	Actor's behavior An new room created An file was uploaded		
Pre-conditions:	Sheet amount must obey business rule.		
Post-conditions:	Sheet is created.		
Main Success Scenario:	<ol style="list-style-type: none"> 1. Creator clicks Create Sheet button 2. Creator is prompted to enter sheet name 3. Creator clicks OK 4. New sheet is created and appended to the right of sheet list. 		

5. Everyone in room will get notification that new sheet is created

Alternative Scenario: N/A

Exceptions: N/A.

Relationships: N/A

Business Rules: N/A.

USE CASE-DELETE SHEET SPECIFICATION

Use-case No.	UC015	Use-case Version	1.0
Use-case Name	Delete Sheet		
Author	NhatNTM		
Date	02/01/2012	Priority	Normal

Actor: Creator

Summary: This use case is about deleting exist sheet in a room.

Goal: Creator can delete.

Triggers: Actor's behavior.

Pre-conditions: N/A

Post-conditions: Sheet is deleted.

Main Success Scenario:

1. Creator clicks Delete button next to sheet name in sheet list from Room Screen
2. Creator confirms that he/she wants to delete sheet.
3. Sheet is deleted from sheet list
4. Everyone in room gets notification.

Alternative Scenario: If creator click Cancel in confirmation box, nothing changes

Exceptions: N/A

Relationships: N/A

Business Rules: N/A

USE CASE-RENAME SHEET SPECIFICATION

Use-case No.	UC016	Use-case Version	1.0
Use-case Name	Rename Sheet		
Author	NhatNTM		
Date	02/02/2012	Priority	Normal

Actor: Creator

Summary: This use case is about renaming exist sheet.

Goal: Creator can change the name of the sheet as they want.

Triggers: Actor's behavior
Pre-conditions: N/A
Post-conditions: Sheet is renamed.

Main Success Scenario:

1. Creator clicks Rename button next to sheet name
2. Creator enters new name and click OK
3. Everyone in room gets notified that sheet has changed its name.

Alternative Scenario: If old and new names are the same, do not notify participants

Exceptions: N/A
Relationships: N/A
Business Rules: N/A

USE CASE-EXPORT SHEET SPECIFICATION

Use-case No.	UC017	Use-case Version	1.0
Use-case Name	Export Sheet		
Author	NhatNTM		
Date	02/01/2012	Priority	Normal

Actor: Participant
Summary: This use case is about exporting existing sheet into xml file or image to download.

Goal: Participant can download the xml for using later or capture the current sheet into image file.

Triggers: Actor's behavior

Pre-conditions: N/A

Post-conditions: Sheet file is converted and downloaded.

Main Success Scenario:

1. Participant clicks download button while open the desired sheet.
2. A save file dialog appears to let that participant choose save location
3. Participant clicks "Save".
4. Participant gets notification that saving operation is completed

Alternative Scenario: N/A

Exceptions: N/A

Relationships: N/A

Business Rules: N/A

USE CASE-IMPORT SHEET SPECIFICATION

Use-case No.	UC018	Use-case Version	1.0
--------------	-------	------------------	-----

Use-case Name	Import Sheet		
Author	NhatNTM		
Date	02/01/2012	Priority	Normal
Actor:	Creator		
Summary:	This use case is about uploading xml file that has been downloaded before into new sheet.		
Goal:	Creator can re-create the sheet that using the history of the xml file they have downloaded.		
Triggers:	Actor's behavior		
Pre-conditions:	N/A		
Post-conditions:	A new sheet with uploaded data is created.		
Main Success Scenario:	<ol style="list-style-type: none"> 1. Creator clicks Upload button from Room Screen 2. A new open file dialog appears for creator to select sheet file 3. Creator selects file and click Open 4. New sheet with uploaded content is created. 5. Everyone in room gets notification that new sheet is created. 		
Alternative Scenario:	N/A		
Exceptions:	Invalid file format: return error message.		
Relationships:	N/A		
Business Rules:	N/A		

System features – Contact Management

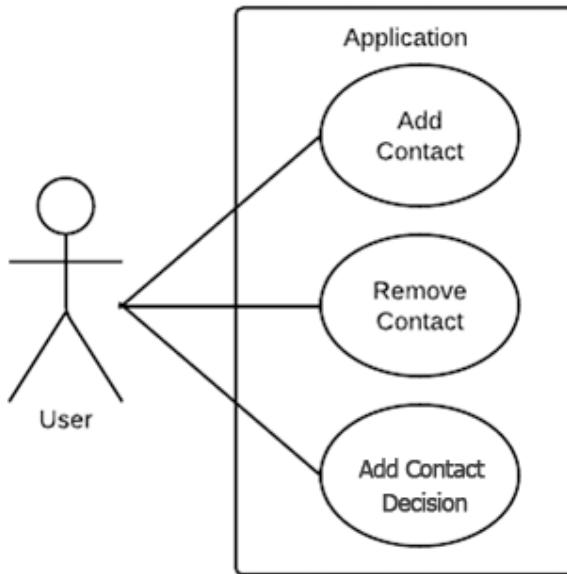


Figure 6 - Contact Management feature use-case diagram

USE CASE-ADD CONTACT SPECIFICATION			
Use-case No.	UC019	Use-case Version	1.0
Use-case Name	Add contact use case		
Author	QuangTV		
Date	31/01/2012	Priority	Normal
Actor:	User		
Summary:	This use case is about Add contact feature		
Goal:	For users to add a contact to their contact list		
Triggers:	Actor's behavior		
Pre-conditions:	N/A		
Post-conditions:	A request is sent to other user		
Main Success Scenario:	<ol style="list-style-type: none"> 1. User clicks Add Contact button. 2. Add contact dialog appears 3. User enters desired contact username, then click Add 4. Added contact appears as pending contact in contact list 5. Added user will get notification immediately or when they log in again. 		
Alternative Scenario:	N/A		
Exceptions:	Contact to be added is not valid or the added contact is already in contact list.		
Relationships:	N/A		
Business Rules:	N/A		

USE CASE-ADD CONTACT DECISION SPECIFICATION			
Use-case No.	UC019	Use-case Version	1.0
Use-case Name	Add contact decision use case		
Author	QuangTV		
Date	31/01/2012	Priority	Normal
Actor:	User		
Summary:	This use case is about Add contact Decision feature		
Goal:	For users to accept or deny a request		
Triggers:	Actor's behavior		
Pre-conditions:	Other user sent request		
Post-conditions:	Contact added		
Main Success Scenario:	<ol style="list-style-type: none"> 1. Other user sent request to this user 2. Immediately, this user get a confirmation box whether he/she wants to accept or not 3. If this user clicks Accept, new contact appears in this user's contact list as accepted. This user will also appear as accepted in the other user's contact list. 		
Alternative Scenario:	<ol style="list-style-type: none"> 1. Other user sent request to this user 2. Immediately, this user get a confirmation box whether he/she wants to accept or not 3. If this user clicks Deny, his/her contact will disappear from the other user's contact list. 		
Exceptions:	N/A		
Relationships:	N/A		
Business Rules:	N/A		

USE CASE-REMOVE CONTACT SPECIFICATION			
Use-case No.	UC020	Use-case Version	1.0
Use-case Name	Remove contact use case		
Author	QuangTV		
Date	31/01/2012	Priority	Normal
Actor:	User		
Summary:	This use case is about Remove contact feature		
Goal:	For users to remove a contact from their contact list		
Triggers:	Actor's behavior		
Pre-conditions:	Contact is in Contact list		
Post-conditions:	Contact was deleted successful		
Main Success Scenario:	<ol style="list-style-type: none"> 1. User clicks delete button next to desired contact name. 		

2. A confirmation box appears
3. User clicks Delete, the selected contact will disappear from contact list

Alternative Scenario:

1. User clicks delete button next to desired contact name.
2. A confirmation box appears
3. User clicks Cancel, nothing changes

Exceptions: N/A

Relationships: N/A

Business Rules: N/A

System features - Sharing

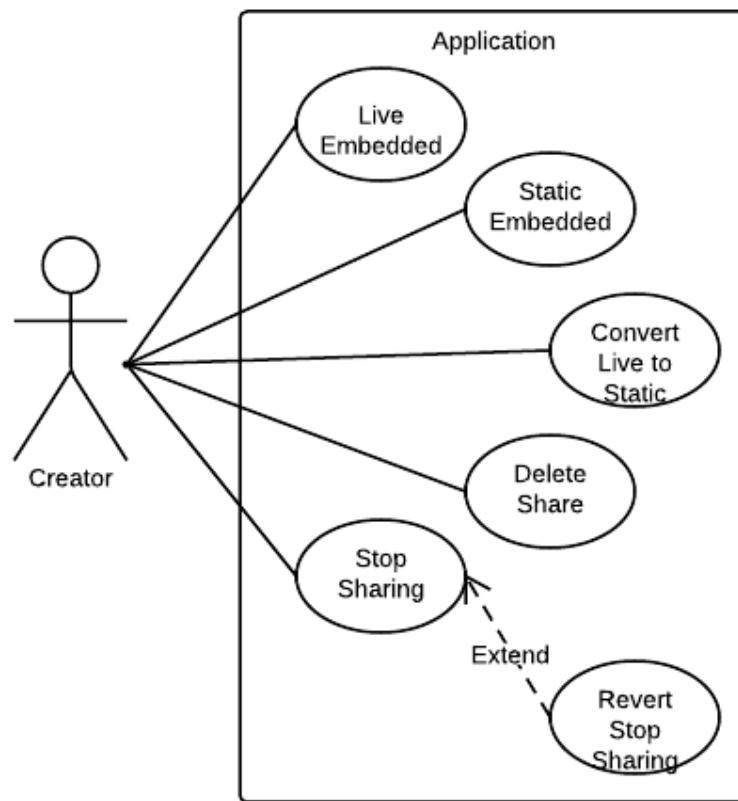


Figure 7- Sharing feature use-case diagram

USE-CASE STATIC EMBEDDED SPECIFICATION						
Use-case No	UC022	Use-case Version	1.0			
Use-case Name	Static embedded					
Author	QuangTV					
Date	31/01/2012	Priority	Normal			
Actor:	Creator					
Summary:	Static share feature					
Goal:	Supports users to share the static content of the sheet to be embedded to other website					
Triggers:	Actor's behavior					
Pre-conditions:	N/A					
Post-conditions:	Sheet is shared successfully					
Main Success Scenario:						
1. Creator click share static button while opening the desired sheet.						
2. New static sheet appears in Share lists in Home Screen and creator gets notified that sharing operation is successful.						
Alternative Scenario:	N/A					
Exceptions:	N/A					
Relationships:	N/A					
Business Rules:	N/A					

USE CASE LIVE EMBEDDED SPECIFICATION						
Use-case No	UC023	Use-case Version	1.0			
Use-case Name	Live embedded					
Author	QuangTV					
Date	31/01/2012	Priority	Normal			
Actor:	Creator					
Summary:	Live share feature					
Goal:	Supports users to share a sheet to other websites					
Triggers:	Actor's behavior					
Pre-conditions:	Sheet is being worked on					
Post-conditions:	The sheet have to be embedded to other site					
Main Success Scenario:						
1. Creator clicks share live button while opening the desired sheet.						
2. New static sheet appears in Share lists in Home Screen and creator gets notified that sharing operation is successful.						
Alternative Scenario:	N/A					
Exceptions:	N/A					
Relationships:	N/A					
Business Rules:	N/A					

USE CASE CONVERT SHARE SHEET SPECIFICATION			
Use-case No	UC024	Use-case Version	1.0
Use-case Name	Convert shared sheet		
Author	QuangTV		
Date	02/02/2012	Priority	Normal
Actor:	Creator		
Summary:	This feature supports users to convert a sheet from live embedded to static embedded		
Goal:	Helps users to convert a sheet from live embedded to static embedded		
Triggers:	Actor's behavior		
Pre-conditions:	Sheet must be live shared		
Post-conditions:	A link to a copied sheet is created as static embedded.		
Main Success Scenario:	<ol style="list-style-type: none"> 1. Creator clicks Convert button next to live sheet name. 2. A confirmation box appears. 3. Creator clicks convert 4. Live sheet becomes static and is moved to Static share list 		
Alternative Scenario:	N/A		
Exceptions:	N/A		
Relationships:	N/A		
Business Rules:	N/A		

USE CASE-STOP SHARING SPECIFICATION			
Use-case No	UC025	Use-case Version	1.0
Use-case Name	Stop Sharing		
Author	ChauLVN		
Date	04/02/2012	Priority	Normal
Actor:	Creator		
Summary:	This use case help creator stop sharing a sheet. This can be undone		
Goal:	Creator successfully stops a share after confirming.		
Triggers:	Actor's behavior		
Pre-conditions:	Share still exists on server.		
Post-conditions:	N/A		
Main Success Scenario:	<ol style="list-style-type: none"> 1. Creator clicks Stop button next to share sheet name. 2. A confirmation box appears. 3. Creator clicks stop 		

4. Sharing is stopped and other site embedding this sheet will get an error message: "This sheet is no longer available"

Alternative Scenario: N/A

Exceptions: N/A

Relationships: N/A

Business Rules: N/A

USE CASE REVERT STOP SHARING SPECIFICATION

Use-case No	UC026	Use-case Version	1.0
Use-case Name	Revert stop sharing		
Author	TungVV		
Date	2/4/2012	Priority	Normal
Actor:	Creator		
Summary:	This feature supports users to continue sharing a sheet that previously stopped		
Goal:	Meet all the user needs listed in user requirement.		
Triggers:	Actor's behavior		
Pre-conditions:	Sheet sharing is currently stopped		
Post-conditions:	Sheet sharing works again		
Main Success Scenario:			
1.	Creator clicks Revert Stop button next to share sheet name.		
2.	A confirmation box appears.		
3.	Creator clicks Revert		
4.	Sharing becomes normal		
Alternative Scenario:	N/A		
Exceptions:	N/A		
Relationships:	N/A		
Business Rules:	N/A		

USE CASE-SHARE DELETE SPECIFICATION

Use-case No	UC027	Use-case Version	1.0
Use-case Name	Share Delete		
Author	NhatNTM		
Date	04/02/2012	Priority	Normal
Actor:	Creator		
Summary:	This use case is about deleting a share. This operation cannot be undone		
Goal:	Creator deletes desired share after confirming.		
Triggers:	Actor's behavior		
Pre-conditions:	Share still exists on server.		

Post-conditions: N/A

Main Success Scenario:

1. Creator clicks Delete button next to share sheet name.
2. A confirmation box appears.
3. Creator clicks Delete
4. Sharing is stopped and other site embedding this sheet will get an error message: "This sheet is no longer available"

Alternative Scenario: N/A

Exceptions: N/A

Relationships: N/A

Business Rules: N/A

System features – Drawing

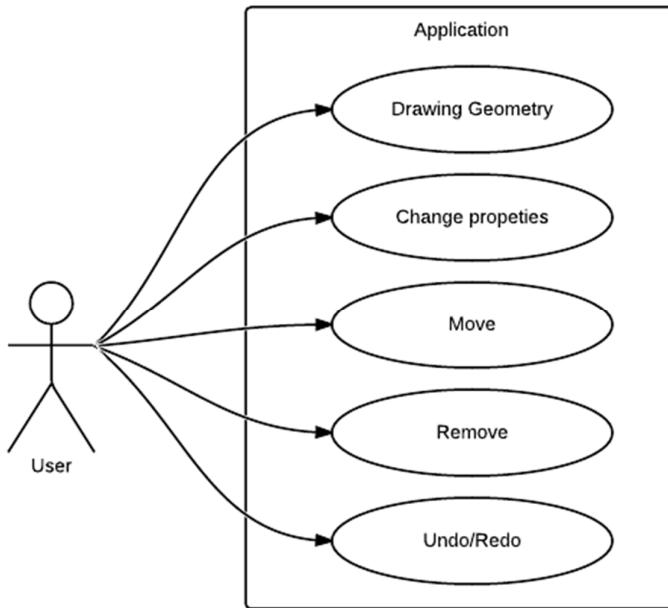


Figure 8 - Drawing feature use-case diagram

USE CASE DRAWING FIGURES SPECIFICATION			
Use-case No	UC028	Use-case Version	2.0
Use-case Name	Drawing figures		
Author	ChauLVN60076		
Date	07/02/2012	Priority	High
Actor:	Broadcaster		
Summary:	This feature supports users to draw all the geometric figures.		
Goal:	Successfully create the chosen figure.		
Triggers:	Actor's behavior		
Pre-conditions:	N/A		
Post-conditions:	The chosen figure is successfully created and displayed.		
Main Success Scenario:	<ol style="list-style-type: none"> 1. Broadcaster selects a tool from toolbox 2. A drawing guide message appears, show how to complete this type of drawing¹ 3. Broadcaster follows the guide and completes drawing the figure 4. A successful message appears 		
Alternative Scenario:	N/A		
Exceptions:	N/A		
Business Rules:	All of the drawing events must be auto saved and can be undone at any time. Other participants can see what is changing.		

¹ See Table 9 – Supported Drawing tools (page A) for completed supported drawing tool.

USE CASE CHANGE PROPERTIES SPECIFICATION			
Use-case No	UC030	Use-case Version	1.0
Use-case Name	Change properties		
Author	ChauLVN60076		
Date	31/01/2012	Priority	Normal
Actor:	Broadcaster		
Summary:	This feature supports users to change the properties of a shape or a text in the sheet on actor's application. Only change from creator and broadcaster will be saved in database.		
Goal:	Helps user changing the property of the selected object (color, style, etc.).		
Triggers:	Actor's behavior.		
Pre-conditions:	An object must be selected		
Post-conditions:	The property is successfully modified and display.		
Main Success Scenario:	<ol style="list-style-type: none"> 1. Broadcaster selects a figure and drawing pad and click edit in property panel. 2. Property panel becomes editable 3. Broadcaster changes desired properties and click Save 4. Selected figure changes visual appearance according to the changes 		
Alternative Scenario:	N/A		
Exceptions:	N/A		
Relationships:	N/A		
Business Rules:	All changes must be auto saved and can be undone at any time. Other participants can see what is changing.		

USE CASE MOVE SPECIFICATION			
Use-case No	UC031	Use-case Version	1.0
Use-case Name	Move		
Author	ChauLVN60076		
Date	31/01/2012	Priority	Normal
Actor:	Broadcaster		
Summary:	This feature supports users to change location of an object in the sheet.		

Goal:	Change location of an object.
Triggers:	Actor's behavior
Pre-conditions:	An object must be selected
Post-conditions:	The object is successfully moved to the new location.
Main Success Scenario:	
1. Broadcaster runs the cursor over a figure, presses and holds the left mouse button, run cursor to a new position, then releases the mouse button.	
2. Selected figure moves along the mouse cursor. Any figure that depends on the selected figure must change its shape as well.	
Alternative Scenario: N/A	
Exceptions:	N/A
Relationships:	N/A
Business Rules:	All changes must be auto saved and can be undone at any time. Other participants can see what is changing.

USE CASE REMOVE SPECIFICATION					
Use-case No	UC032	Use-case Version	1.0		
Use-case Name	Remove				
Author	ChauLVN60076				
Date	31/01/2012	Priority	Normal		
Actor:	Broadcaster				
Summary:	This feature supports users to delete an object in the sheet.				
Goal:	The object is successfully removed.				
Triggers:	Actor's behavior				
Pre-conditions:	An object must be selected				
Post-conditions:	The object is successfully removed.				
Main Success Scenario:					
1. Broadcaster selects a figure and click Delete button in property panel or click Del on the keyboard to remove it.					
2. The selected figure is removed from drawing pad. Any figure that depends on the selected figure must be removed as well.					
Alternative Scenario: N/A					
Exceptions:	N/A				
Relationships:	N/A				
Business Rules:	All changes must be auto saved and can be undone at any time. Other participants can see what is changing.				

USE CASE UNDO/REDO SPECIFICATION			
Use-case No	UC034	Use-case Version	1.0
Use-case Name	Undo/Redo		
Author	ChauLVN60076		
Date	31/01/2012	Priority	Normal
Actor:	Broadcaster.		
Summary:	This feature supports users to undo/redo their actions in the sheet.		
Goal:	User can undo and redo their action from sheet's history.		
Triggers:	User's action		
Pre-conditions:	Must have actions to undo, or previously undone action to redo.		
Post-conditions:	The action is successfully undone or redone.		
Main Success Scenario:	<ol style="list-style-type: none"> 1. Broadcaster clicks undo button (redo button) to undo (redo, respectively) the last change from the opening sheet. 2. The change is undone or redone immediately. 		
Alternative Scenario:	N/A		
Exceptions:	N/A		
Relationships:	N/A		
Business Rules:	All other participants in the room can see the change		

Software System Attributes

Reliability

The system shall never crash or hang, other than as the result of an operating system or network error.

- Mean Time Between Failures (MTBF): The acceptable failure is once a year. The failure should not be because of the database, or else the data may be lost and cannot be recovered.
- Mean Time To Repair (MTTR): When the failure occurs, it should take as least time as possible to repair. The acceptable mean time for a particular failure must be least than 8 hours.
- Maximum Bugs or Defect Rate: 1-2bugs/KLOC

Availability

The server shall be working 24/7. When the system go in under-maintenance, the page will display that 'the website is not available that that moment. Pleased come back in some hours.' and advise for the users some webpage related.

Security

There are no specific security and privacy requirements.

Maintainability

All code shall be fully documented. All program files shall include comments concerning authorship and date of last change.

The code shall be modular to permit future modifications.

Portability

The software shall be designed to run in web browser that support Silverlight

Performance

This is the system's performance characteristics.

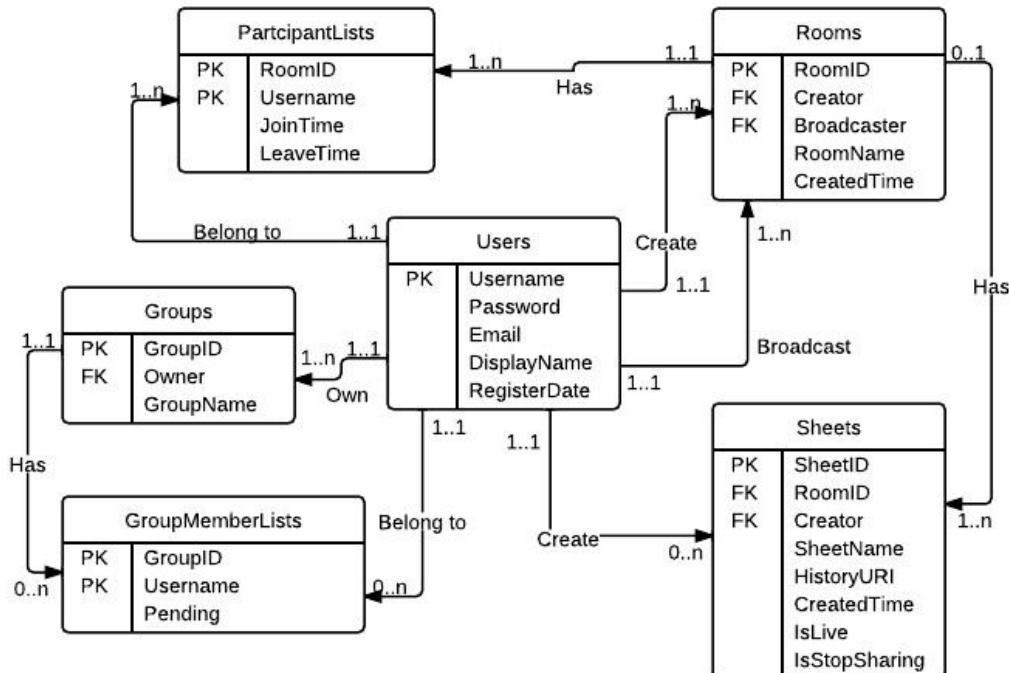
- Throughput: 1000 transactions per second.
- Capacity: 200 customers a time.
- Response time for a transaction:
 - Average: 1 seconds
 - Maximum: 5 seconds

Entity Relationship Diagram

This diagram describes all the entities in this system and their import properties.

NOTE:

For more detail about Entity Relationship, see Database Design (page 88)



Software Design Description

Design Overview

This section provides a high-level description of the goals of the architecture supported by the system and architectural styles, components that have been selected to best achieve the use cases from SRS document. This document defines and guides the design criteria, technical and domain standards.

System Architectural Design

Choice of System Architecture

This system will apply 3-tier combined with service-oriented architecture. The system has a clear separation between presentation tier, application tier and data tier. Additionally, application tier (or business logic provider) is delivered through a combination of web services. The following diagram depicts the overview concept of this system:

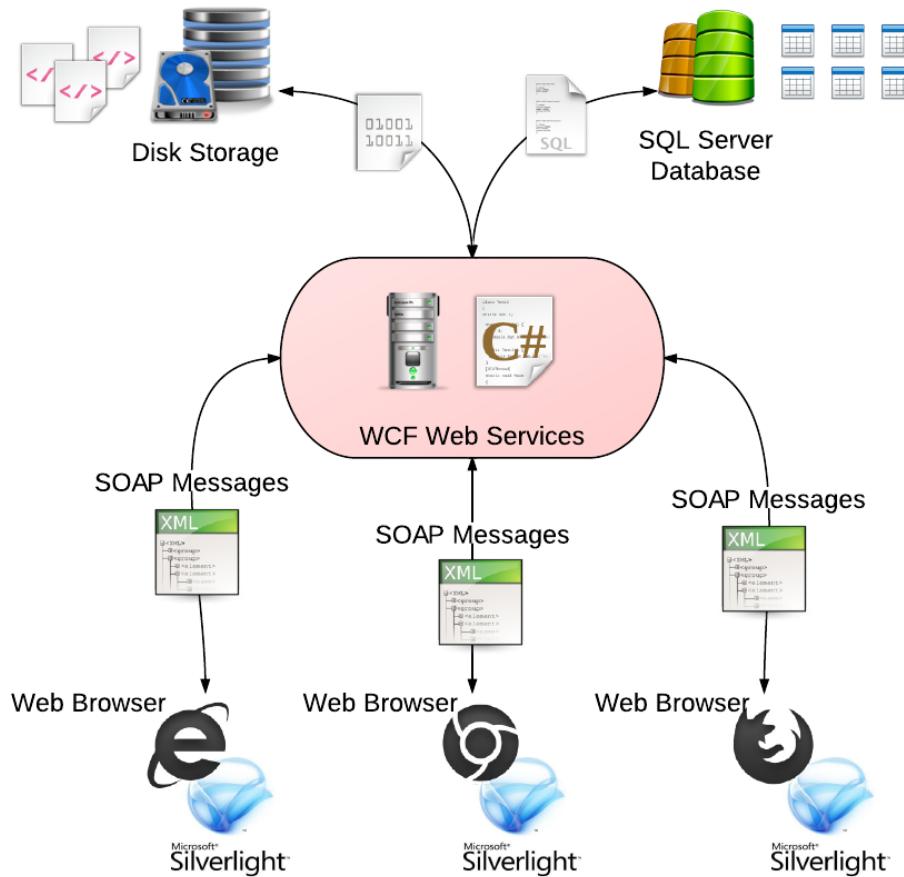


Figure 9 - Pythagore Architecture

- **Presentation tier** includes any web browser that can run Silverlight application (example: Google Chrome browser with Silverlight runtime plugin²). This tier takes the responsibility to render a User Graphical Interface for end-users to interact and it will generate corresponding requests to Application tier.
- **Application tier** includes a collection of WCF³ web services written in .net language that will process request from presentation tier, make response back to that tier, and interact with Data tier to complete a data-related command.
- **Data tier** includes Database server that run a Microsoft SQL Express Server to process data-related commands (create, read, update, and delete data), and disk storage to store persistent data files.

In the scope of this project, user can only interact with Silverlight application. However, thanks to the extensibility and portability of XML documents, the presentation tier can be extended to run on, theoretically, any platform, for instance, mobile phones or HTML5 webpages.

Discussion of Alternative Designs

N/A

Description of System Interface

The system primarily interfaces with other system using XML documents exchange. Other system can use the web service component as a public API.

²Silverlight plugin can be found here: <http://www.microsoft.com/getsilverlight>

³ WCF is stand for Windows Communication Foundation – an application programming interface (API) in the .NET Framework for building connected, service-oriented applications

Component Diagram

The following diagram shows all the parts of the design of Pythagore system. It provides the visualization of the high-level structure of the system and the service behaviors that those pieces provide and consume through interfaces

Pythagore system consists of 4 important components:

1. **Common Library**: Provides geometric and mathematical logic, can run in both service (Full .net framework runtime) and client (Silverlight runtime).
2. **Pythagore Service**: Provides business logic and service interfaces
3. **Silverlight Library**: Provides Visual implementation for Silverlight runtime
4. **Silverlight Client**: Provides User interface and is the consumer of web services

Detail explanations for each component are conducted in next sections

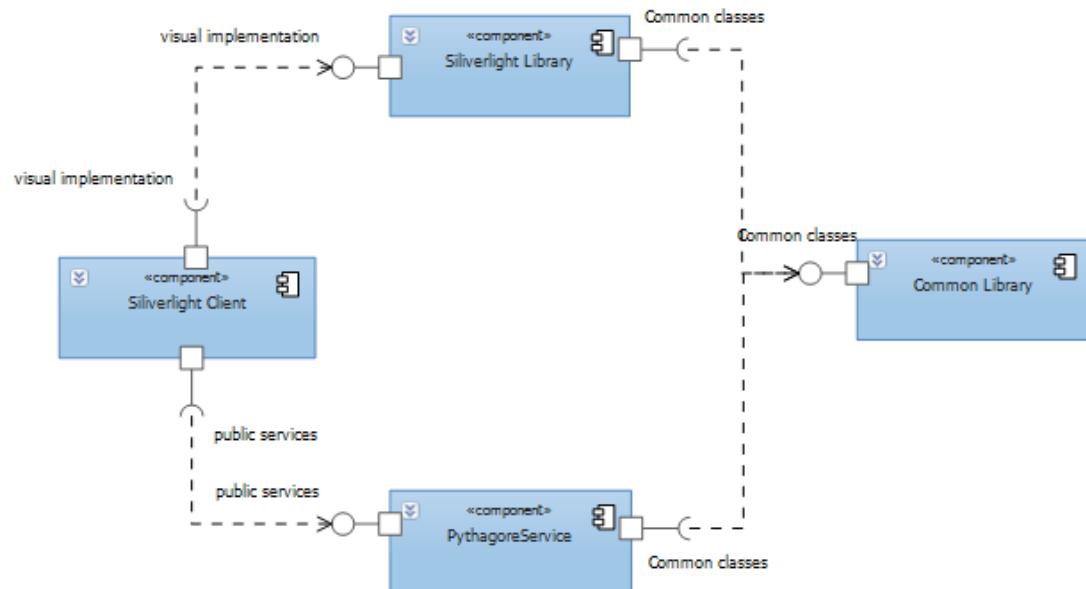


Figure 10 - Pythagore Component Diagram

Detail Description of Components

Common Library

Common Library component provides basic functionalities that can be run on all other components. Common Library must be a Portable Class Library⁴ type of assembly.

Class diagram

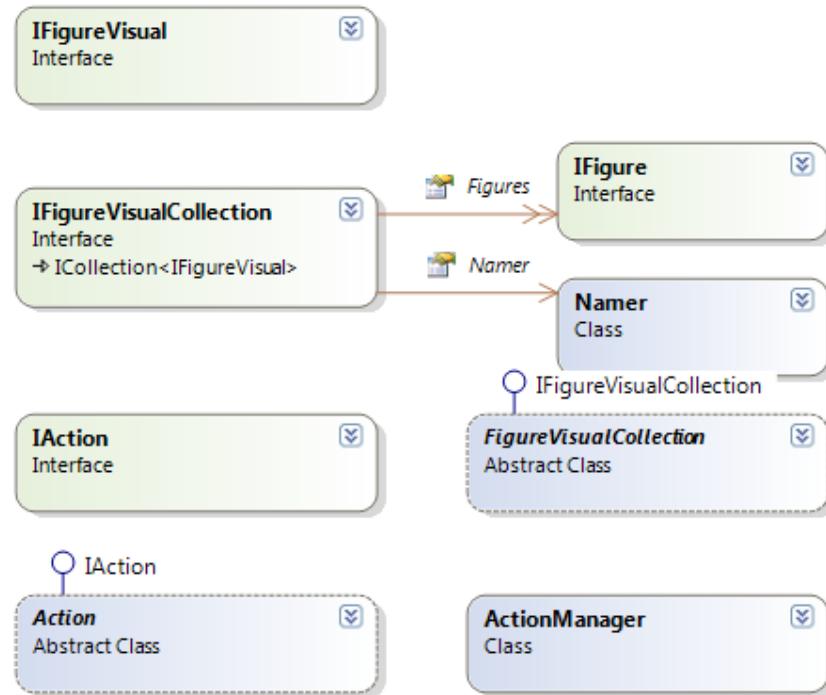


Figure 11 - Overview class diagram for Common library

Class Diagram Explanation

This component should consist of 2 main namespaces: Common.Actions and Common.Figures.

⁴The Portable Class Library project enables you to write and build managed assemblies that work on more than one .NET Framework platform.

Namespace: Common.Actions

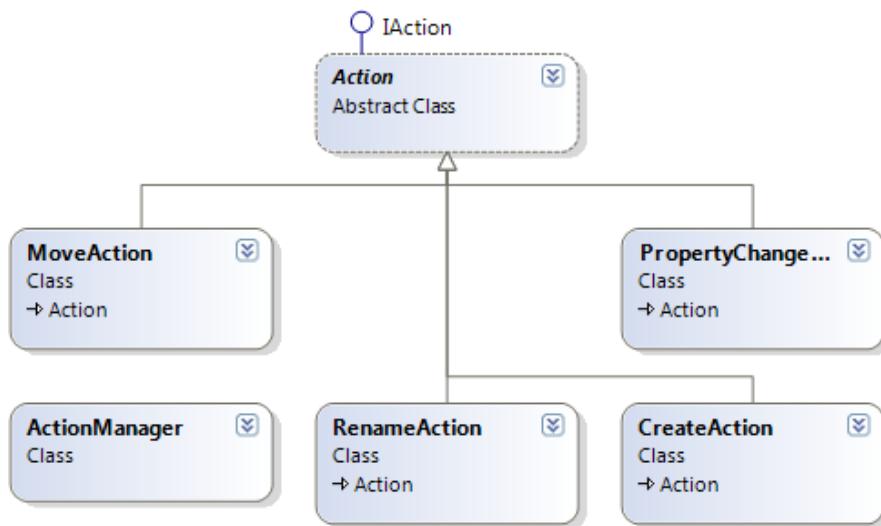


Figure 12 - Common.Actions class diagram

Interface: Common.Actions.IAction

Interface for `Action` class.

Inherited by `Common.Actions.Action`

Functions

void	Do ()
	Do the action.
void	Undo ()
	Undo the action.

Properties

bool	CanUndo [get]
	Check if an action is undoable.
bool	CanDo [get]
	Check if an action is doable.

Class: Common.Actions.Action

Inherits `Common.Actions.IAction`

Inherited by `Common.Actions.CreateAction`, `Common.Actions.MoveAction`, and `Common.PropertyChange`

Functions

abstract void	Do ()
	Do the action.
abstract void	Undo ()
	Undo the action.

Properties

int	ExecutionCount [get, set]
-----	----------------------------------

Action count variable, if ExecutionCount > 0, can Undo.

Class: Common.Actions.CreateAction

Support create method for an action

Inherits **Common.Actions.Action**

Functions

	CreateAction (IFigure figure, IFigureVisualCollection collection)
	Constructor with 2 params.
override void	Do ()
	If action is doable, do the action.
override void	Undo ()
	If action is undoable, undo the action.

Properties

IFigure	Figure [get, set]
	Get Set method for Figure.
IFigureVisualCollection	Collection [get, set]
	Get Set method for Collection.

Class: Common.Actions.MoveAction

Supports moving method for action

Inherits **Common.Actions.Action**

Functions

	MoveAction (IFigure figure, Vector distance, bool done=false)
	Move figure to another location.
override void	Do ()
	If action is doable, do the action.
override void	Undo ()
	If action is undoable, undo the action.

Common.Actions.PropertyChangeAction

Inherits **Common.Actions.Action**.

Public Member Functions

	PropertyChangeAction (Geometry2D figure, string propertyName, object oldValue, object newValue, bool executed=false)
	Constructor with 2 params.
override void	Do ()
	Do the action.
override void	Undo ()
	Undo the action.

Common.Actions.RenameAction

Inherits **Common.Actions.Action**.

Public Member Functions

	RenameAction (IFigure figure, string newName)
override void	Do () Do the action.
override void	Undo () Undo the action.

Class: Common.Actions.ActionManager

Support managing actions

Functions

void	AppendAction (IAction action) Merging actions if possible before adding actions to history before performing them.
void	Undo () Support returning to the next state.
void	Redo () Support returning to the previous state.
bool	ExecuteCommand (string command) Execute a command. If the command cannot be executed, return false.

Properties

IFigureVisualCollection	Collection [get, set] Return a collection of visual figure.
IHistory	History [get, set] Get and Set history.
bool	CanUndo [get] Check if an action can be undone
bool	CanRedo [get] Check if an action can be redone.

Events

ActionEventHandler	OnAdding Event that invoked when a new action is adding.
ActionEventHandler	OnAdded Event that invoked when a new action is added.
ActionEventHandler	OnUndoing Event that invoked when action is undoing.
ActionEventHandler	OnUndone Event that invoked when action is undone.
ActionEventHandler	OnRedoing Event that invoked when action is redoing.
ActionEventHandler	OnRedone Event that invoked when action is redone.

HistoryEventHandler **OnChanged**

Event that invoked when something is changed.

Namespace: Common.Figures

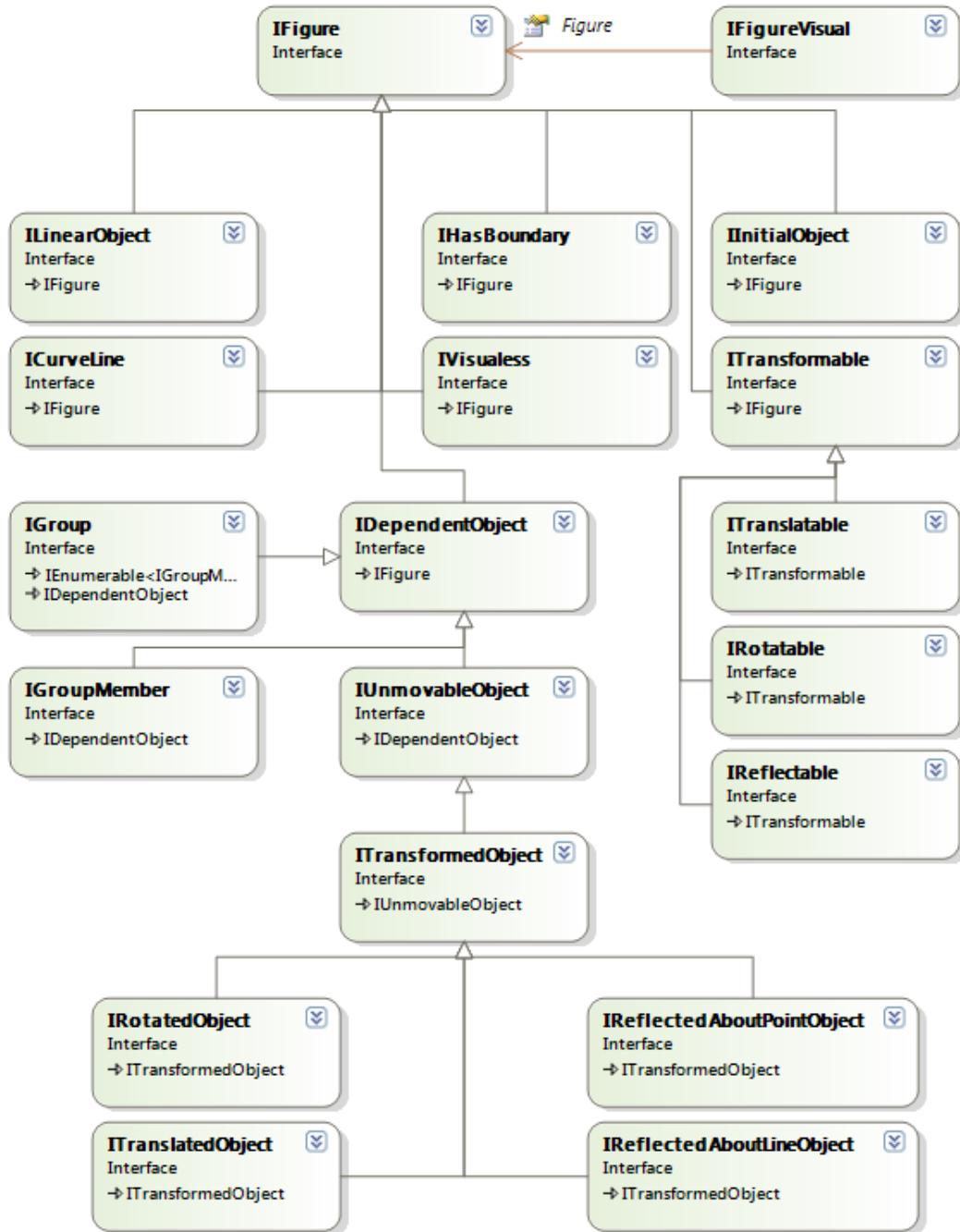


Figure 13 - Common.Figures namespace class diagram – part 1 of 2

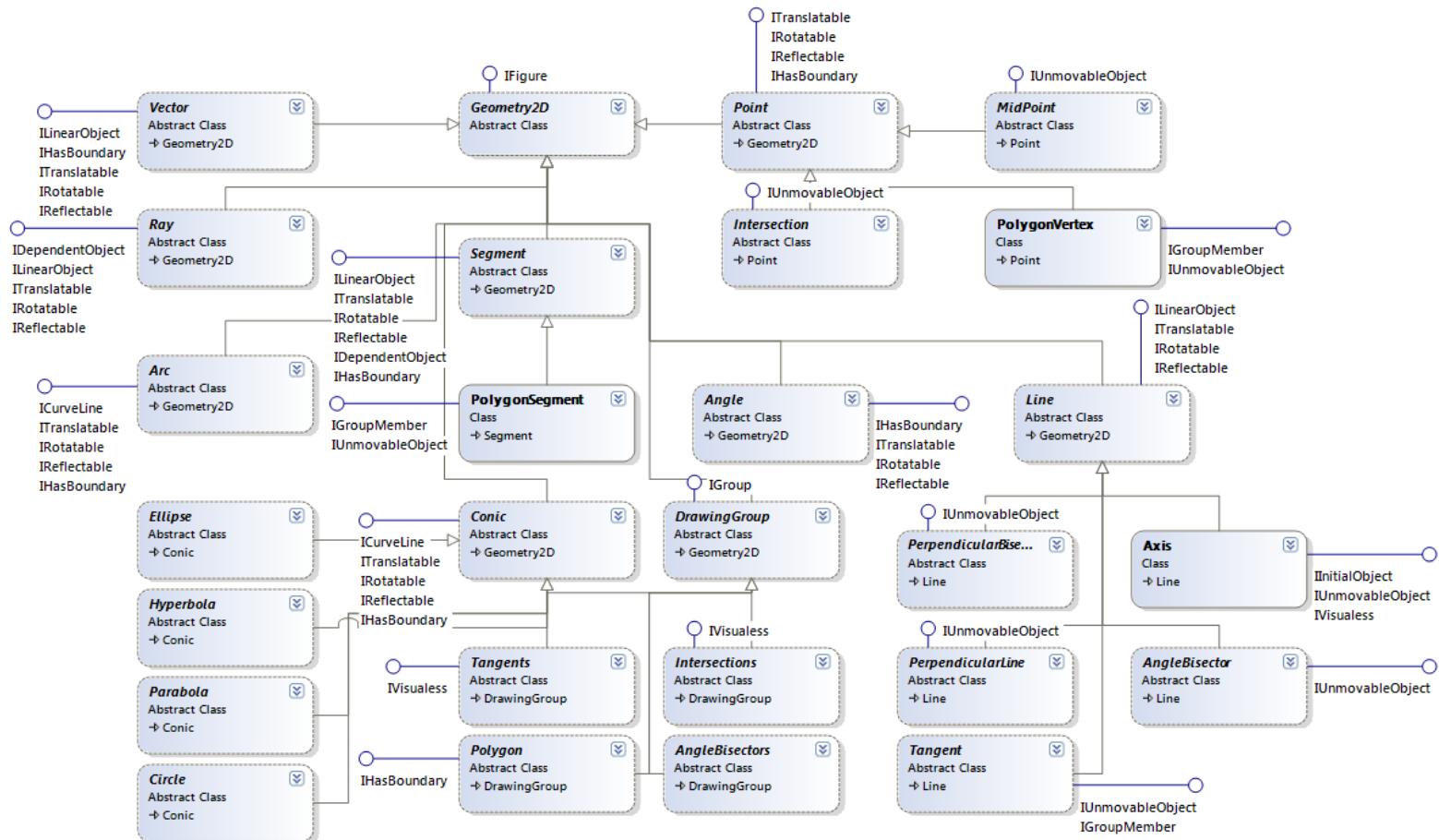


Figure 14 - Common.Figures namespace class diagram – part 2 of 2

Common.IFigure Interface Reference

Parent of everything

Public Member Functions

void	Move (Vector distance)
List< IFigure >	Remove ()
string	DefinitionText (bool byName=false)
Queue< IFigure >	FindDependent ()

Properties

string	ID [get, set] Unique ID.
string	Name [get, set] Object name.
string	FullName [get, set] Object full name.
string	Title [get] Object description.
VisualDetail	VisualDetail [get, set] Appearance information
FigureChangeState	Changes [get] Flag for change states

Events

FigureChangeHandler	OnVisualChanged
FigureChangeHandler	OnMoved
FigureChangeHandler	OnShapeChanged
FigureChangeHandler	OnNameChanged
FigureChangeHandler	OnRemoved

Common.IDependentObject

Object that depends on others

Inherits **Common.IFigure**

Properties

List< IFigure >	DependentGeometries [get]
------------------------	----------------------------------

Common.ICurveLine

Parent of curved figures

Inherits **Common.IFigure**

Public Member Functions

bool	IsOnCurve (Point p) Test whether a point is on this curved figure
List< Point >	PointsOfTangency (Point m) Get a list of tangency points from a point

Properties

Equation	Equation [get]
-----------------	-----------------------

Point **CurveCenter** [get]

Common.ILinearObject

Parents of linear figures.

Inherits **Common.IFigure**

Public Member Functions

bool	IsOnLine (Point p)
Test whether a point is on this linear figure	

Properties

Point	Anchor [get] An unique point that this linear figure go through
double	Magnitude [get] A numeric value for linear relative length
Vector	DirectionVector [get] Direction vector

Common.IGroup

Presentation of a group of objects that go together

Inherits **Common.IDependentObject**

Properties

List<	IGroupMember	> Members [get]
-------	---------------------	------------------------

Common.ITransformedObject

Objects that is transformed from another object

Inherits **Common.IUnmovableObject**

Properties

IFigure	Origin [get, set]
----------------	--------------------------

Common.IRotatedObject

Objects that is rotated from another object.

Inherits **Common.ITransformedObject**.

Properties

Point	Center [get, set]
AngleMagnitude	Angle [get, set]

Common.IReflectedAboutPointObject

Objects that is reflected from another object.

Inherits **Common.ITransformedObject**.

Properties

Point	Mirror [get, set]
--------------	--------------------------

Common.IReflectedAboutLineObject

Objects that is reflected from another object

Inherits **Common.ITransformedObject**.

Properties

ILinearObject	Mirror [get, set]
----------------------	--------------------------

Common.ITranslatedObject

Objects that is translated from another object

Inherits **Common.ITransformedObject**.

Properties

Vector	Distance [get, set]
---------------	----------------------------

Common.ITransformable

Objects that can be transformed. Inherits **Common.IFigure**.

Common.IReflectable

Objects that can be reflected.

Inherits **Common.ITransformable**.

Public Member Functions

IReflectedAboutPointObject	CreateReflected (Point mirror)
IReflectedAboutLineObject	CreateReflected (ILinearObject mirror)

Common.ITranslatable

Objects that can be translated.

Inherits **Common.ITransformable**.

Public Member Functions

ITranslatedObject	CreateTranslated (Vector distance)
--------------------------	---

Common.IRotatable

Objects that can be rotated.

Inherits **Common.ITransformable**.

Public Member Functions

IRotatedObject	CreateRotated (Point center, AngleMagnitude angle)
-----------------------	--

Other interfaces

Table 6 – Other interfaces

Interface	summary
Common.IInitialObject	objects to be initialized when application starts (eg: O _x , O _y)
Common.IVisualless	Object that doesn't has an actual appearance
Common.IHasBoundary	object that has a boundary (for example: circle, segment)
Common.IGroupMember	Object that is a member of a group.

Class:Common.Figures.Segment

Data definition for **Segment**

Inherits **Common.Figures.Geometry2D**, **Common.ILinearObject**, **Common.ITranslatable**, **Common.IRotatable**, **Common.IReflectable**, and **Common.IDependentObject**

Properties

virtual Point	End1 [get, set]
	Return start point of the segment.
virtual Point	End2 [get, set]
	Return end point of the segment.
double	Length [get]
	Return the length of the segment.
virtual List< IFigure >	DependentGeometries [get]
	Return list of figure of this geometry.
Point	Anchor [get]
	Return the anchor point of segment.
double	Magnitude [get]
	Return the magnitude of the segment. In segment, magnitude = length.
Vector	DirectionVector [get]
	Return the vector of the segment.

Subclasses

Class Name	Class Description
SegmentReflectedAboutLine	Support drawing reflection of a segment through a line.
SegmentReflectedAboutPoint	Support drawing reflection of a segment through a point.
SegmentRotated	Support rotating a segment.
SegmentTranslated	Support translating a segment.
Segment2Point	Implement method for Segment created by 2 points.

Class: Common.Figures.Vector

Data definition for **Vector**

Inherits **Common.Figures.Geometry2D**, **Common.ILinearObject**, **Common.IHasBoundary**, **Common.ITranslatable**, **Common.IRotatable**, and **Common.IReflectable**

Public member functions

bool	IsSameDirection (Vector u)
	Return true if this vector has same direction with passed vector.
Vector	Negate ()
	Return the opposed direct vector of current vector.
Vector	CreateUnitVector ()
	Create a vector from each unit.
bool	IsParallel (Vector a)
	Return true if this vector parallel with passed vector.

Vector	MakeConsistent () Return the consistent vector.
Properties	
abstract double	X [get, set] Return X value of the vector.
abstract double	Y [get, set] Return Y value of the vector.
abstract Point	Tail [get, set] Return the point the vector end.
abstract Point	Head [get, set] Return the point the vector start.
double	Length [get] Return the length of the vector.
virtual AngleMagnitude	AngleFromOx [get] Return the angle between this vector and Ox Axis.
Point	Anchor [get] Return the anchor of the vector.
double	Magnitude [get] Return the magnitude of the vector.
Vector	DirectionVector [get] Return this vector.
Subclasses	
Class Name	Class Description
VectorReflectedAboutLine	Support drawing reflection of a vector through a line.
VectorReflectedAboutPoint	Support drawing reflection of a vector through a point.
VectorRotated	Support rotating a vector.
Vector2Point	Vector created by 2 points: tail & head.

Class: Common.Figures.Ray

Data definition for **Ray**.

Inherits **Common.Figures.Geometry2D**, **Common.IDependentObject**, **Common.ILinearObject**, **Common.ITranslatable**, **Common.IRotatable**, and **Common.IReflectable**

Properties

virtual Point	EndPoint [get, set] Return the point where the ray end.
virtual internal Point	PassThroughPoint [get, set]
Point	Anchor [get] Return the anchor point of the ray.
abstract double	Magnitude [get] Return the magnitude of the ray.
abstract Vector	DirectionVector [get, set] Return the direction vector of the ray.

abstract List< IFigure >	DependentGeometries [get]
Return the list of figures of the ray.	

Subclasses

Class Name	Class Description
RayReflectedAboutLine	Support drawing reflection of a Ray through a line.
RayReflectedAboutPoint	Support drawing reflection of a Ray through a point.
RayRotated	Support rotating a Ray.
RayTranslated	Support translating a Ray.
Ray2Point	Support method for create a Ray from 2 Points.
RayPointVector	Support method for create a Ray from a Point and a Vector.

Class: Common.Figures.Line

Data definition for Line.

Inherits **Common.Figures.Geometry2D**, **Common.ILinearObject**, **Common.ITranslatable**, **Common.IRotatable**, and **Common.IReflectable**

Properties

abstract Point	Anchor [get]
Return the anchor point of the line.	
abstract double	Magnitude [get]
Return the magnitude of the line, for comparing.	
abstract Vector	DirectionVector [get, set]
Return the direction vector of the line.	
virtual Equation	Equation [get, set]
Return the equation of the line.	

Class: Common.Figures.Concrete.AngleBisector

Data definition for Angle Bisector

Inherits **Common.Figures.Line**, and **Common.IUnmovableObject**

Properties

virtual Point	Vertex [get, set]
Return the point where angle bisector is on.	
Vector	VectorA [get, set]
Return the first vector of the angle bisector.	
Vector	VectorB [get, set]
Return the second vector of the angle bisector.	
abstract List< IFigure >	DependentGeometries [get]
Return the list of figure of the angle bisector.	
override Point	Anchor [get]
Return the anchor of the angle bisector.	
override double	Magnitude [get]
Return the magniture of the angle bisector.	

override **Equation** **Equation** [get]

Return the equation of the angle bisector.

Class: Common.Figures.Tangent

Support method for create **Tangent** of other figures.

Inherits **Common.Figures.Line**, **Common.IUnmovableObject**, and **Common.IGroupMember**

Properties

internal **Conic** **Conic** [get, set]

override **Vector** **DirectionVector** [get, set]

Return the direction vector of the line.

override **Point** **Anchor** [get]

Return the anchor point of the line.

override double **Magnitude** [get]

Return the magnitude of the line, for comparing.

override **Equation** **Equation** [get]

Return the equation of the line.

List< **IFigure** > **DependentGeometries** [get]

Line's Subclasses

Class Name	Class Description
AngleBisector2Line	Support method for create an Angle with 2 lines.
AngleBisector3Point	Support method for create an Angle with 3 Points.
ConicAxis	Support method for create Axis for Conic .
Line2Points	Support method for create a Line with 2 Points.
LineWithEquation	Support method for create a Line with an Equation .
LineWithLineAndPoint	Support method for create a Line with a Point and a Line .
LineWithVectorAndPoint	Support method for create a Line with a Point and a Vector .
PerpendicularBisector	Data definition for Perpendicular Bisector.
PerpendicularBisector2Points	Support method for create an Perpendicular Bisector of 2 Points.
PerpendicularBisectorOfSegment	Support method for create an Perpendicular Bisector of a Segment .
PerpendicularLine	Data definition for PerpendicularLine.
PerpendicularPointLine	Support method for drawing a new perpendicular line through a point and a line.
PerpendicularPointVetor	Support method for drawing a perpendicular line using a point and a vector.
LineReflectedAboutLine	Support drawing reflection of a Line through a line.
LineReflectedAboutPoint	Support drawing reflection of a Line through a point.
LineRotated	Support rotating a Line.
LineTranslated	Support translating a Line.

Common.Figures.Arc

Data definition for **Arc**

Inherits **Common.Figures.Geometry2D**, **Common.ICurveLine**, **Common.ITranslatable**, **Common.IRotatable**, **Common.IReflectable**, and **Common.IHasBoundary**

Properties

virtual Point	Center [get, set] Return the center point of the arc.
virtual Point	Start [get, set] Return the start point of the arc.
virtual Point	End [get, set] Return the end point of the arc.
virtual Angle	Angle [get, set] Return the angle of the arc.
virtual double	Radius [get, set] Return the radius from center point of the arc.
virtual double	Length [get, set] Return the length of the arc.
Equation	Equation [get] Return the equation of the arc.

Common.Figures.Concrete.CircularArc

Support method for create a Circular **Arc**.

Inherits **Common.Figures.Arc**, and **Common.IDependentObject**

Public Member Functions

	CircularArc (Point center, Point p1, Point p2)
	create a circular arc from 3 point. The radius is the distance from center to the first point.

Properties

override Point	End [get, set] Return the end point of the arc.
override Angle	Angle [get, set] Return the angle of the arc.
override double	Length [get, set] Return the length of the arc.
override double	Radius [get, set] Return the radius from center point of the arc.

Common.Figures.Concrete.CircumcircularArc

Support method for create a Circumcircular **Arc**.

Inherits **Common.Figures.Arc**, and **Common.IDependentObject**

Public Member Functions

	CircumcircularArc (Point start, Point midPoint, Point end)
--	---

Properties

	override Point	Start [get, set] Return the start point of the arc.
	override Point	End [get, set] Return the end point of the arc.
	Point	MidPoint [get] override double Length [get, set] Return the length of the arc.
	override Angle	Angle [get, set] angle create by vector OA and OC with O is the center of the arc
	override Point	Center [get, set] Return the center point of the arc.
	override double	Radius [get, set] Return the radius from center point of the arc.

Common.Figures.Concrete.Semicircle

Support method for create a **Semicircle Arc**.

Inherits **Common.Figures.Arc**, and **Common.IDependentObject**

Public Member Functions

		Semicircle (Point a, Point b) create a semicircle from 2 points
Properties		
	override Point	Center [get, set] Return the center point of the arc.
	override double	Radius [get, set] Return the radius from center point of the arc.
	override double	Length [get, set] Return the length of the arc.
	override Angle	Angle [get, set] Return the angle of the arc.

Subclasses

Class Name	Class Description
ArcReflectedAboutLine	Support drawing reflection of an Arc through a line.
ArcReflectedAboutPoint	Support drawing reflection of an Arc through a point.
ArcRotated	Support rotating an Arc.
ArcTranslated	Support translating an Arc.

Common.Figures.Angle

Data definition for **Angle**

Inherits **Common.Figures.Geometry2D**, **Common.IHasBoundary**, **Common.ITranslatable**, **Common.IRotatable**, and **Common.IReflectable**

Properties

virtual Point	Vertex [get, set] Return the point where the angle position is.
virtual Vector	VectorA [get, set] Return first vector that create the angle.
virtual Vector	VectorB [get, set] Return second vector that create the angle.
virtual AngleMagnitude	Magnitude [get, set] Return the magnitude of the angle.
double	Degree [get]
double	Radian [get]
override string	Title [get] Return the name of the angle.

Subclasses

Class Name	Class Description
AngleReflectedAboutLine	Support drawing reflection of an Angle through a line.
AngleReflectedAboutPoint	Support drawing reflection of an Angle through a point.
AngleRotated	Support rotating an Angle.
AngleTranslated	Support translating an Angle.
Angle2Lines	Draw an angle when 2 lines are specified.
Angle2Points	Draw an angle when 2 points and magnitude value is given.
Angle3Points	Draw an angle when 3 points is specified.

Common.Figures.Polygon

Inherits **Common.Figures.DrawingGroup**, and **Common.IHasBoundary**

Properties

abstract List< Point >	Vertices [get]
List< PolygonSegment >	Segments [get, set]
abstract int	VertexCount [get]
virtual double	Area [get] Return the area value of the polygon.

Subclasses

Class Name	Class Description
PolygonReflectedAboutLine	Support drawing reflection of a Polygon through a line.
PolygonReflectedAboutPoint	Support drawing reflection of a Polygon through a point.
PolygonRotated	Support rotating a Polygon .
PolygonTranslated	Support translating a Polygon .
PolygonByPoint	Implement method for define a Polygon with a given Point.
RegularPolygon	Implement method for define a regular Polygon .

Common.Figures.Point

Data definition for **Point**

Inherits **Common.Figures.Geometry2D**, **Common.ITranslatable**, **Common.IRotatable**, **Common.IReflectable**, and **Common.IHasBoundary**

Properties

abstract double	X [get, set] Return the X value of the point.
abstract double	Y [get, set] Return the Y value of the point.

Subclasses

Class name	Description
Common.Figures.Concrete.PointOnObject	Implement method for define a point on an object.
Common.Figures.Concrete.PointReflectedAboutLine	Support drawing reflection of a Point through a Line.
Common.Figures.Concrete.PointReflectedAboutPoint	Support drawing reflection of a Point through a Point.
Common.Figures.Concrete.PointXY	Class for point that create by providing x-, y-coordinate.
Common.Figures.Concrete.PointOnLine	Class for point that attach on line.
Common.Figures.Concrete.PointOnCurve	Class for point that attach on curve.
Common.Figures.Concrete.PolygonVertex	Support method for create a Polygon Vertex.

Common.Figures.Intersection

Data definition for **Intersection**

Inherits **Common.Figures.Point**, and **Common.IUnmovableObject**

Public Attributes

IFigure	Object1
IFigure	Object2

Subclasses

Class name	Description
Common.Figures.Concrete.Intersection2Conics	Intersection between two conics.
Common.Figures.Concrete.IntersectionLineConic	Intersection between a line and a conic.
Common.Figures.Concrete.IntersectionLineLine	Intersection between two lines.

Common.Figures.MidPoint

Data definition for Midpoint

Inherits **Common.Figures.Point**, and **Common.IUnmovableObject**

Properties

override double	X [get, set] Return X value of the point.
override double	Y [get, set] Return Y value of the point.
abstract Point	Head [get, set] Return the first point of the segment where midpoint is on.
abstract Point	Tail [get, set]

Return the second point of the segment where midpoint is on.

Subclasses

Class name	Description
Common.Figures.Concrete.Midpoint2Point	Define midpoint from 2 points.
Common.Figures.Concrete.MidpointSegment	Define midpoint for a segment.

Common.Figures.Conic

Data definition for **Conic**.

Inherits **Common.Figures.Geometry2D**, **Common.ICurveLine**, **Common.ITranslatable**, **Common.IRotatable**, **Common.IReflectable**, and **Common.IHasBoundary**

Public Member Functions

List< Point >	PointsOfTangency (Point m)
----------------------	-----------------------------------

Return a list of tangent points.

Properties

virtual Equation	Equation [get, set]
-------------------------	----------------------------

Return the equation of the conic.

Common.Figures.Circle

Abstract and Factory to create Circles

Inherits **Common.Figures.Conic**

Properties

abstract double	Radius [get, set]
-----------------	--------------------------

Return the radius.

abstract Point	Center [get, set]
-----------------------	--------------------------

Return the center point.

override Equation	Equation [get]
--------------------------	-----------------------

Return the name of the circle.

Subclasses

Class name	Description
Common.Figures.Circle3Point	Implement method for Circle created by 3 points.
Common.Figures.Concrete.Circle2Point	Implement method for Circle created by 2 points.
Common.Figures.Concrete.CircleReflected AboutLine	Implement method for Circle created by reflected with a line.
Common.Figures.Concrete.CircleReflected AboutPoint	Implement method for Circle created by reflected with a point.
Common.Figures.Concrete.CircleRotated	Implement method for Circle created by rotated.
Common.Figures.Concrete.CircleWithEquation	Implement method for Circle created by an Equation.
Common.Figures.Concrete.CircleWithRadius	Implement method for Circle created with a given Radius.

Common.Figures.Ellipse

Data definition for **Ellipse**.

Inherits **Common.Figures.Conic**

Properties

virtual Point	F1 [get, set] Return the first focus point of ellipse.
virtual Point	F2 [get, set] Return the second focus point of ellipse.
virtual Point	Center [get] Return the center point of ellipse.
virtual AngleMagnitude	RotatingAngle [get] Return the angle between vector F1,F2 and Ox.
virtual double	MajorRadius [get, set] Return the major radius of ellipse.
virtual double	MinorRadius [get, set] Return the minor radius of ellipse.
Line	MajorSemiAxis [get] Return the line where major axis of ellipse is on.
Line	MinorSemiAxis [get] Return the line where minor axis of ellipse is on.
virtual double	FocusLength [get] Return the focus length of ellipse.
double	Eccentricity [get] The eccentricity of ellipse.
override Equation	Equation [get, set] Return the equation of ellipse.

Subclasses

Class name	Description
Common.Figures.Concrete.Ellipse3Point	Implement method for Ellipse created by 3 points.
Common.Figures.Concrete.EllipseReflected AboutLine	Implement method for Ellipse created by reflecting with a line.
Common.Figures.Concrete.EllipseReflected AboutPoint	Implement method for Ellipse created by reflecting with a point.
Common.Figures.Concrete.EllipseRotated	Implement method for Ellipse created by rotating.
Common.Figures.Concrete.EllipseTranslate d	Implement method for Ellipse created by translating.

Common.Figures.Hyperbola

Data definition for **Hyperbola**.

Inherits **Common.Figures.Conic**.

Properties

virtual Point	F1 [get, set] Return the first focal point of hyperbola.
virtual Point	F2 [get, set] Return the second focal point of hyperbola.
virtual Point	Vertex1 [get, set] Return the first intersection between focal line and hyperbola.
virtual Point	Vertex2 [get, set] Return the second intersection between focal line and hyperbola.
virtual Point	Center [get] Return the midpoint of two focal points of hyperbola.
virtual AngleMagnitude	RotatingAngle [get] Return the angle between vector F1,F2 and Ox.
virtual double	MajorRadius [get, set] Return the major radius of hyperbola.
virtual double	MinorRadius [get, set] Return the minor radius of hyperbola.
Line	MajorAxis [get] Return the line where major axis of hyperbola is on.
Line	MinorAxis [get] Return the line where minor axis of hyperbola is on.
virtual double	FocusLength [get] Return the length between two focal points.
double	Eccentricity [get] Return the eccentricity of hyperbola.
override Equation	Equation [get, set] Return the equation of hyperbola.

Subclasses

Class name	Description
Common.Figures.Concrete.Hyperbola3Point	Implement method for Hyperbola created by 3 points.
Common.Figures.Concrete.HyperbolaReflectedAboutOutline	Implement method for Hyperbola created by reflecting with a line.
Common.Figures.Concrete.HyperbolaReflectedAboutPoint	Implement method for Hyperbola created by reflecting with a point.
Common.Figures.Concrete.HyperbolaRotated	Implement method for Hyperbola created by rotating.
Common.Figures.Concrete.HyperbolaTranslated	Implement method for Hyperbola created by translating.

Common.Figures.Parabola

Data definition for **Parabola**.

Inherits **Common.Figures.Conic**.

Properties

virtual Point	Focus [get, set] Return the focus point of the parabola.
virtual ILinearObject	Directrix [get, set] Return the directrix of the parabola.
override Equation	Equation [get, set] Return the equation of the parabola.
override Point	CurveCenter [get]
virtual Line	Axis [get] Return the axis of the parabola.
override string	Title [get] Return the name of parabola.

Subclasses

Class name	Description
Common.Figures.Concrete.ParabolaFocusDirectrix	Implement method for a Parabola created by a Directrix and a Point .
Common.Figures.Concrete.ParabolaReflectedAboutLine	Implement method for Parabola created by reflected with a line.
Common.Figures.Concrete.ParabolaReflectedAboutPoint	Implement method for Parabola created by reflected with a point.
Common.Figures.Concrete.ParabolaRotated	Implement method for Parabola created by rotated.
Common.Figures.Concrete.ParabolaWithEquation	Implement method for a Parabola created by an Equation .

Silverlight Library

Class diagram

The following diagram provides overall view of Silverlight Library Component:

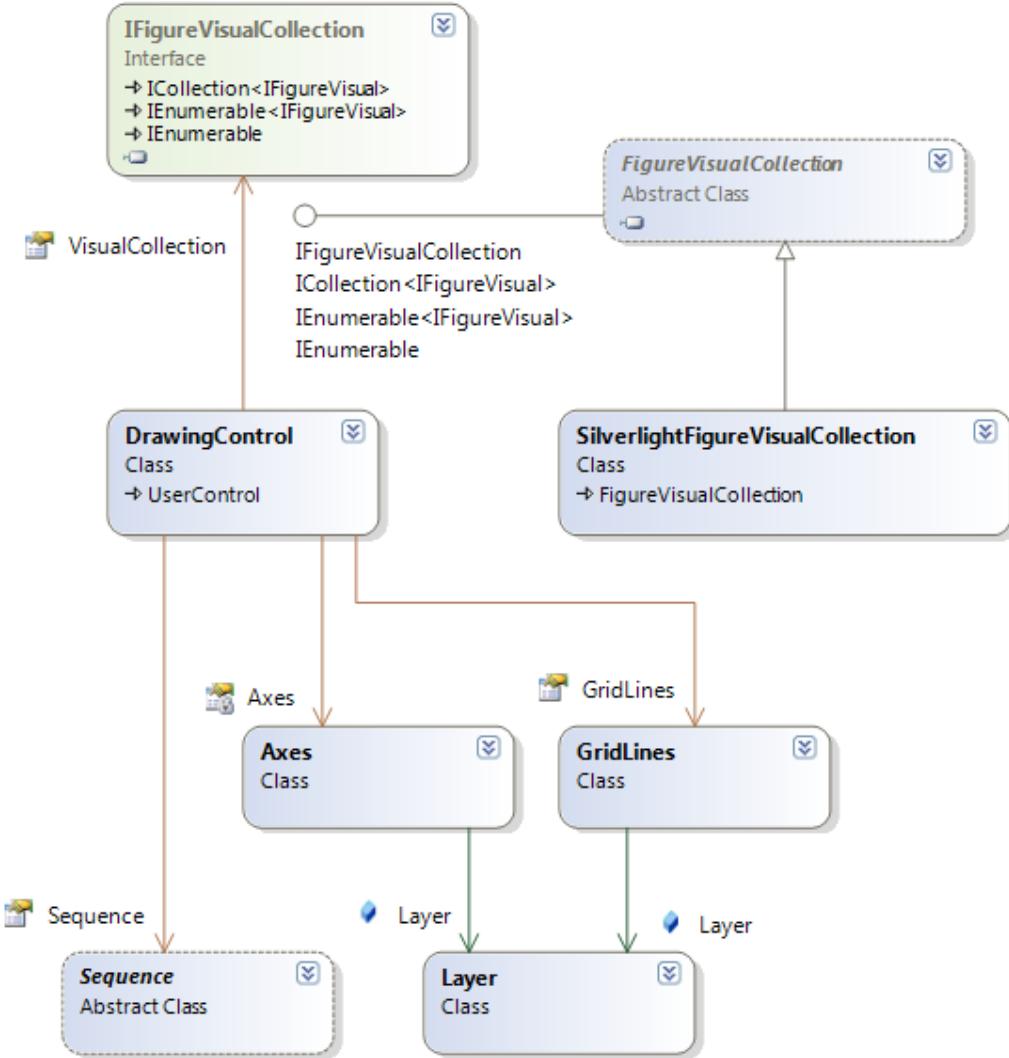


Figure 15 - Silverlight Library Component Overview

Class diagram explanation

This component has 3 main namespaces: Controls, DrawingSequences and Visuals.

SilverlightLibrary.Visuals Namespace

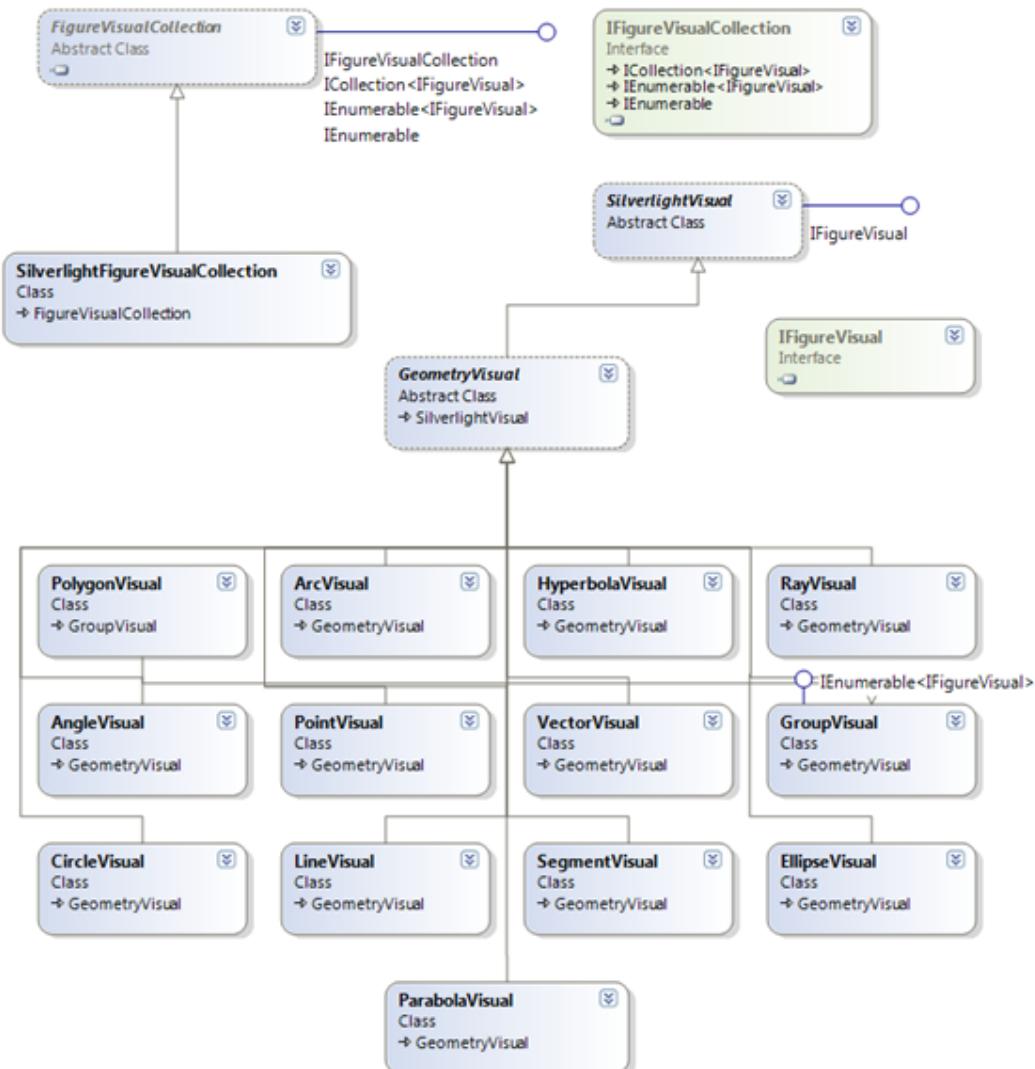


Figure 16-SilverlightLibrary.Visual Namespace

SilverlightLibrary.Visual.SilverlightFigureVisualCollection

Collection of **IFigureVisual** implemented for Silverlight.

Inherits **Common.Figures.FigureVisualCollection**

Public Member Functions

override void	Draw () Invoke all IFigureVisual to draw itself
override void	Add (IFigure figure) Add a new Figure to the collection (a new visual will be created)
override bool	Remove (IFigureVisual visual)

SilverlightLibrary.Visual.SilverlightVisual

Support drawing in general.

Inherits **Common.Figures.IFigureVisual**

Public Member Functions

void	Draw ()
	Execute the figure action.
virtual void	Clear ()
	Clear this graph from the graph Layer.

Properties

Layer	DrawingLayer [get, set]
Path	Shape [get, set]
bool	IsDragging [get, set] Whether the figure is being dragged or not.
bool	IsOutOfBounds [get, set]
bool	IsHover [get, set] Whether the figure is hover or not.
bool	IsSelected [get, set] Whether the figure is being selected or not.
bool	AllowMove [get, set]
abstract IFigure	Figure [get] Underlying figure

Events

RenderEventHandler	OnRendering
RenderEventHandler	OnRendered
DragDropEventHandler	OnDrop
EventHandler	OnSelectedChanged

SilverlightLibrary.Visual.GeometryVisual

Support drawing Geometry objects

Inherits **SilverlightLibrary.Visual.SilverlightVisual**

Properties

Geometry2D	Geometry [get, set] Get the underlaying figure as Geometry.
VisualDetail	Detail [get] Get, set the color of geometry visual detail.
TextBlock	Label [get, set] Get the label object.
abstract Point	LabelPosition [get] Get the position for label.
ToolTip	ToolTip [get] Get the tooltip object.
bool	Selectable [get, set]

Subclasses

Subclass	Description
PointVisual	Support drawing a point on drawing pad
SegmentVisual	Support drawing a segment on drawing pad
CircleVisual	Support drawing a circle on drawing pad
EllipseVisual	Support drawing an ellipse on drawing pad
PolygonVisual	Support drawing a polygon on drawing pad
ArcVisual	Support drawing an arc on drawing pad
VectorVisual	Support drawing a vector on drawing pad
LineVisual	Support drawing a line on drawing pad
RayVisual	Support drawing a ray on drawing pad
ParabolaVisual	Support drawing a parabola on drawing pad
HyperbolaVisual	Support drawing a hyperbola on drawing pad
AngleVisual	Support drawing an angle on drawing pad
GroupVisual	Support drawing a group of geometries on drawing pad

SilverlightLibrary.DrawingSequences Namespace

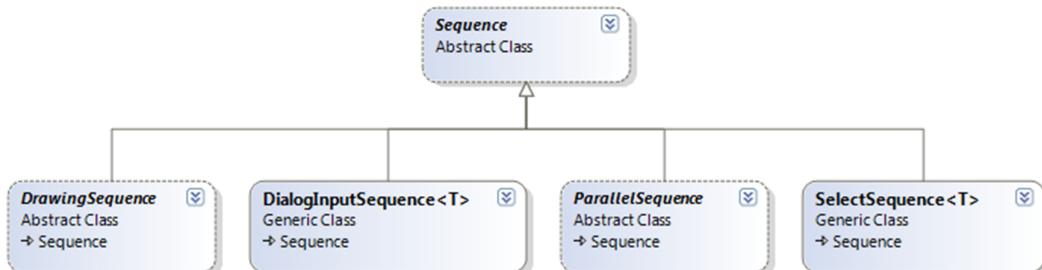


Figure 17 - SilverlightLibrary.DrawingSequences class diagram

Class: SilverlightLibrary.DrawingSequences.Sequence

Support user to complete a desired behavior, step by step.

Public Member Functions

abstract void	Stop ()
	Stop this sequence immediately.
void	Start ()
	Invoke this sequence to start.

Properties

abstract IFigure	Result [get]
	Result of this sequence as it completes.
abstract string	HelpText [get]
	Message that will show to help user to complete this sequence.
abstract string	SuccessMessage [get]

	Message that will show when this sequence successfully completes.
virtual string	FailedMessage [get] Message that will show when this sequence fails.
virtual string	CancelMessage [get] Message that will show when user cancels this sequence.
Events	
SequenceEventHandler	OnStart Event before sequence started.
SequenceEventHandler	OnFinished Event after sequence completion.
SequenceEventHandler	OnPreview Event on sequence prevision.

SilverlightLibrary.DrawingSequences.DrawingSequence

Support drawing behavior.

Inherits **SilverlightLibrary.DrawingSequences.Sequence**

Properties

DrawingControl	Control [get, set] Current DrawingControl being drawn on.
-----------------------	---

SilverlightLibrary.DrawingSequences.SelectSequence<T>

Support selecting behavior.

Inherits **SilverlightLibrary.DrawingSequences.Sequence**

Public Member Functions

SelectSequence (DrawingControl control, Action< Sequence, SequenceEventArgs > onStart=null, Action< Sequence, SequenceEventArgs > onFinish=null, Action< Sequence, SequenceEventArgs > onPreview=null, bool autoStart=false)
--

SilverlightLibrary.DrawingSequences.DialogInputSequence<T>

Support user input behavior.

Inherits **SilverlightLibrary.DrawingSequences.Sequence**.

Public Member Functions

DialogInputSequence (T dialog, Action< Sequence, SequenceEventArgs > onStart=null, Action< Sequence, SequenceEventArgs > onFinished=null, Action< Sequence, SequenceEventArgs > onPreview=null, bool autoStart=false)

Properties

T	Dialog [get, set]
---	--------------------------

SilverlightLibrary.DrawingSequences.PointDrawingSequence

Support drawing behavior for points.

Inherits [SilverlightLibrary.DrawingSequences.DrawingSequence](#)

Public Member Functions

```
PointDrawingSequence (DrawingControl control,
Action< Sequence, SequenceEventArgs > onStart=null,
Action< Sequence, SequenceEventArgs > onFinish=null,
Action< Sequence, SequenceEventArgs > onPreview=null, bool
autoStart=false)
```

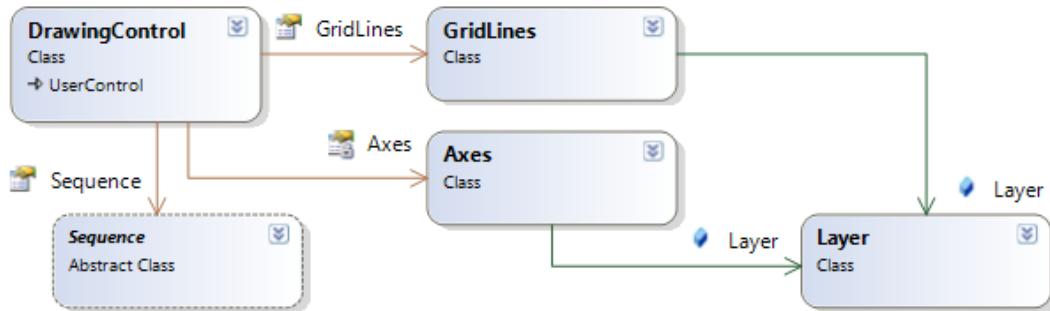
Subclasses:

Class	Description
Angles	
Angle2LinesDrawingSequence	Support drawing behavior for angle between 2 lines
Angle3PointsDrawingSequence	Support drawing behavior for angle between 3 points
AngleWithMagnitudeDrawingSequence	Support drawing behavior for angle with a given magnitude
Angle Bisector	
AngleBisector2LinesDrawingSequence	Support drawing behavior for angle bisector between 2 lines
AngleBisector3PointsDrawingSequence	Support drawing behavior for angle bisector between 3 points
Arc	
CircumcircularArcDrawingSequence	Support drawing behavior for circumcircular arc
SemicircleDrawingSequence	Support drawing behavior for semicircle
CircularArcDrawingSequence	Support drawing behavior for circular arc
Circle	
Circle2PointDrawingSequence	Support drawing behavior for circle with a point on circle and another as center
Circle3PointDrawingSequence	Support drawing behavior for circle with 3 points on circle
CircleWithRadiusDrawingSequence	Support drawing behavior for circle with center and radius
Other conic	
Ellipse3PointDrawingSequence	Support drawing behavior for ellipse with 2 foci and a point on ellipse
Hyperbola3PointDrawingSequence	Support drawing behavior for hyperbola with 2 foci and a point on hyperbola

ParabolaPointDirectrixDrawingSequence	Support drawing behavior for parabola with focus and a directrix
Point	
IntersectionDrawingSequence	Support drawing behavior for intersection of 2 objects
MidPointDrawingSequence	Support drawing behavior for midpoint of 2 points
PointOnLineDrawingSequence	Support drawing behavior for point on line
PointOnCurveDrawingSequence	Support drawing behavior for point on curve
Line	
Line2PointDrawingSequence	Support drawing behavior for a line go through 2 points
LineParallelDrawingSequence	Support drawing behavior for a line go through a point and parallel to a linear object
PerpendicularLinePointDrawingSequence	Support drawing behavior for line that perpendicular to another
Perpendicular Bisector	
PerpendicularBisector2PointDrawingSequence	Support drawing behavior for perpendicular bisector of 2 points
PerpendicularBisectorSegmentDrawingSequence	Support drawing behavior for perpendicular bisector of a segment
Polygon	
RegularPolygonDrawingSequence	Support drawing behavior for a regular polygon
PolygonDrawingSequence	Support drawing behavior for a irregular polygon
Other linear objects	
Ray2PointDrawingSequence	Support drawing behavior for a ray with an end and go through a point
Vector2PointDrawingSequence	Support drawing behavior for a vector from a point to another
Transformation	
RelectionAboutLineDrawingSequence	Support drawing behavior for reflection of an object about a line
ReflectionAboutPointDrawingSequence	Support drawing behavior for reflection of an object about a point
RotationDrawingSequence	Support drawing behavior for rotation of an object
TranslationDrawingSequence	Support drawing behavior for translation of an object

SilverlightLibrary.ControlsNamespace

This namespace provides a user control with complete GUI and API for interacting with a sheet



SilverlightLibrary.Controls.DrawingControl

User control that support user to interact with figures on a drawing pad.

Public Member Functions

void	Zoom (double delta)
WriteableBitmap	CaptureScreen ()
void	Move (Vector distance) Move the drawing pad.
void	StopSequence ()
void	Draw < TFigure > () Start a new drawing sequence to create a new figure.
void	Update () Redraw all elements.

Properties

ActionManager	ActionManager [get, set] Get, set action manager for this control.
GridLines	GridLines [get, set] Get GridLines element.
Sequence	Sequence [get, set] Get, set the current sequence in this control, null if no sequence is running.
bool	IsSequenceContinuous [get, set]
IFigureVisualCollection	VisualCollection [get, set] Get, set the collection of IVisualFigure.
Axes	Axes [get, set] Get Axes element.
Vector	OriginVector [get, set] Get, set the distance from center to origin point.

bool	AllowMouseZoom [get, set]
bool	AllowMouseMove [get, set]
bool	AllowMove [get, set]
double	PixelPerUnit [get, set]
	Get, set the distance (in pixel) between 2 closest unit point.

Events

SequenceEventHandler	OnStartSequence
SequenceEventHandler	OnFinishSequence
DrawingControlEventHandler	OnSelectedFigureChanged

SilverlightLibrary.DrawingPad.GridLines

Drawing logic for grid lines

Public Member Functions

void	Draw ()
	Drawing logic.
void	Move ()
void	Clear ()
	Remove all elements.

Properties

bool	Visible [get, set]
	get, set visibility of grid lines
double	Thickness [get, set]
	Thickness of grid lines.
Brush	Stroke [get, set]
	Brush stroke of grid lines.
int	HorizontalCount [get, set]
	Get the number of horizontal grid lines.
int	VerticalCount [get, set]
	Get the number of vertical grid lines.
double	Gap [get, set]
	Get and set the distance between two closest grid lines.

SilverlightLibrary.DrawingPad.Axes

Drawing logic for axes

Public Member Functions

	Axes (Layer layer)
void	Draw ()
	Draw this Element.
void	Clear ()
	Remove the two axes. Need redrawing.

Properties

double	MinX [get, set]
double	MaxX [get, set]
double	MinY [get, set]
double	MaxY [get, set]
bool	OuterFrameVisible [get, set] Get and set visibility of OuterFrame.
bool	RulerVisible [get, set] Get and set visibility of the rulers on each axis.
bool	AutoSwitchRuler [get, set] Auto switch ruler on each axis to outer frame if the axis is invisible.
bool	VerticalAxisVisible [get, set] Get the visibility of vertical axis.
bool	HorizontalAxisVisible [get, set] Get the visibility of horizontal axis.
double	VerticalLength [get, set] The length of vertical axis (or Height)
double	HorizontalLength [get, set] The length of horizontal axis (or Width)
bool	Visible [get, set]
double	Thickness [get, set] Thickness of each axis.
Brush	Stroke [get, set] Stroke of each axis.
double	Step [get] Distance in unit between two closest grid.

SilverlightLibrary.DrawingPad.Layer

Public Member Functions

Point	GetPixelPoint (Common.Figures.Point unitPoint) Convert an unit point to pixel point.
Point	GetPixelPoint (double x, double y)
Common.Figures.Point	GetUnitPoint (Point pixelPoint) Convert a pixel point to unit point.
Common.Figures.Point	GetUnitPoint (double x, double y)
void	Add (UIElement ui) Add an UIElement onto this layer.
void	Clear () Clear all elements on this Layer .
void	Update () Force an update on layout.
int	ChangeZIndex (int newIndex) Change z-index of a layer.

Properties

	double	PixelPerUnit [get, set] Distance (in pixel) between 2 closest unit points.
	Vector	OriginVector [get, set] distance from the center to the origin point
	double	ActualHeight [get]
	double	ActualWidth [get]
	Point	Center [get] Return the center point of this Layer .
Common.Figures.Point		TopLeft [get] Return the top left in unit point.
Common.Figures.Point		BottomRight [get] Return the bottom right in unit point.
	bool	Updated [get, set] true if the layer has a property changed, false otherwise

Pythagore Service

Class diagram

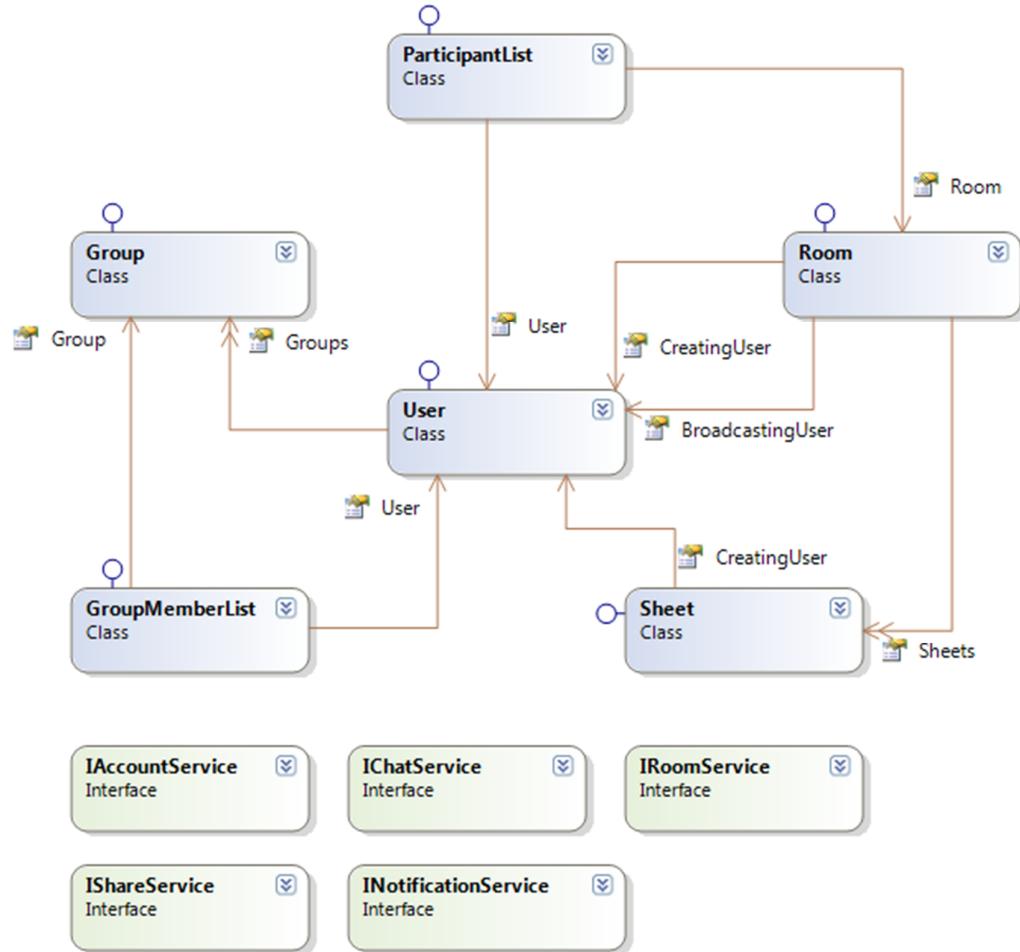


Figure 18 - Pythagore Services class diagram

Class diagram explanation

This component should consist of 2 main namespaces: Entities and Services

Namespace: PythagoreService.Entities

Class: PythagoreService.Entities.ParticipantList

Properties

string	RoomID [get, set]
string	Username [get, set]
System.Nullable<System.DateTime>	JoinTime [get, set]
System.Nullable<System.DateTime>	LeaveTime [get, set]
User	User [get, set]
Room	Room [get, set]

Class: PythagoreService.Entities.Group

Properties

string	GroupID [get, set]
string	Owner [get, set]
string	GroupName [get, set]
EntitySet< GroupMemberList >	GroupMemberLists [get, set]
User	User [get, set]

Class: PythagoreService.Entities.GroupMemberList

Properties

string	GroupID [get, set]
string	Username [get, set]
System.Nullable< bool >	IsPending [get, set]
User	User [get, set]
Group	Group [get, set]

Class: PythagoreService.Entities.Room

Properties

string	RoomID [get, set]
string	Creator [get, set]
string	Broadcaster [get, set]
string	RoomName [get, set]
System.Nullable< System.DateTime >	CreatedTime [get, set]
EntitySet< ParticipantList >	ParticipantLists [get, set]
EntitySet< Sheet >	Sheets [get, set]
User	BroadcastingUser [get, set]
User	CreatingUser [get, set]

Class: PythagoreService.Entities.Sheet

Properties

string	SheetID [get, set]
string	RoomID [get, set]
string	Creator [get, set]
string	SheetName [get, set]
string	HistoryURI [get, set]
System.Nullable< System.DateTime >	CreatedTime [get, set]
bool	IsLive [get, set]

	bool	IsStopSharing [get, set]
	Room	Room [get, set]
	User	CreatingUser [get, set]

Class: PythagoreService.Entities.User

Properties

	string	Username [get, set]
	string	Password [get, set]
	string	Email [get, set]
	string	DisplayName [get, set]
	string	Token [get, set]
	System.DateTime	RegisterDate [get, set]
EntitySet< GroupMemberList >		GroupMemberLists [get, set]
EntitySet< Group >		Groups [get, set]
EntitySet< ParticipantList >		ParticipantLists [get, set]
EntitySet< Room >		BroadcastingRooms [get, set]
EntitySet< Room >		CreatedRooms [get, set]
EntitySet< Sheet >		Sheets [get, set]

Namespace: PythagoreService.Services

PythagoreService.IRoomService Interface Reference

Interface for Room Service.

Public Member Functions

long	SaveState (string token, string sheetID, string events)
	Save data for a room to server.
Room	CreateRoom (string token, string roomName)
	Create new room.
bool	DeleteRoom (string token, string roomID)
	Delete a room with their sheets.
bool	LeaveRoom (string token, string roomID)
	Leave room.
bool	RemoveParticipant (string token, string roomID, string username)
	Kick other user out of room without knowing their token.
bool	InviteUser (string token, string roomID, string[] users)
	Invite other user to join a room.
bool	JoinRoom (string token, string roomID)

		Support loggedin user to join a room.
Sheet	AddSheet (string token, string roomID, string sheetName)	Add new sheet.
bool	RenameSheet (string token, string sheetID, string newSheetName)	Rename a sheet.
bool	RemoveSheet (string token, string sheetID)	Remove a sheet from Room.
Stream	ExportSheet (string token, string sheetID, byte type)	Export sheet to client as XML or image format.
string	ImportSheet (string token, byte[] fileStream, string roomID, string sheetName)	Import a XML file as sheet.
bool	GrantPermission (string token, string roomID, string username)	Grant permission for participant to draw.
bool	TakePermission (string token, string roomID)	Return permission to draw back to owner.
bool	RaiseHand (string token, string roomID)	Support participant to ask for permission to draw.
List< Room >	GetOwnRoom (string token)	To Get the list of room created by current user.
List< Room >	GetInvitedRoom (string token)	To Get the list of room that the current user is invited.
List< Room >	GetInvitedRoomNotJoin (string token)	To Get the list of room that the current user is invited but not join room yet.
bool	RenameRoom (string token, string roomID, string newName)	Rename room.
List< Sheet >	GetSheetList (string token, string roomID)	

PythagoreService.IAccountService Interface Reference

Interface for Authentication Service.

Public Member Functions

User	Login (string username, string password)	
		Support users to login the system.
void	LogOut (string token)	
		Support users to log out the system.
bool	Register (User user, string password)	
		Support users to regist in the system.
bool	CreateGroup (string token, Group group)	
		Create new group.
bool	DeleteGroup (string token, string groupID)	

		Delete a group.
bool	RenameGroup (string token, string groupID, string newGroupName)	Support renaming a group.
List< GroupMemberList >	GetContacts (string token)	View pending contact.
User	AddContact (string token, string username, string groupID)	Add new contact.
bool	DeleteContact (string token, string username)	Delete a contact.
bool	ResponseContact (string token, string username, bool response)	Response to another user after add contact.
bool	ChangeGroup (string token, string username, string newGroupID)	Change participants between groups.
List< GroupDetail >	GetGroups (string owner)	Get groups.
User	GetInfo (string token, string username)	Get contact information.
User	UpdateInfo (string token, User user)	Update user info.
bool	ChangePassword (string token, string oldPassword, string newPassword)	Support changing user's password.

Interface: PythagoreService.Services.IChatService

Interface for Chat Service.

Public Member Functions

bool	SendData (string token, string message)
	Send new message to server.

PythagoreService.IShareService Interface Reference

Interface for Share Service.

Public Member Functions

bool	CreateStaticShare (string token, string sheetID)
	Support creating a static-shared sheet.
bool	SetLive (string token, string sheetID)
	Set a sheet to live shrare.
bool	StopShare (string token, string sheetID)
	Stop sharing a sheet.
bool	DeleteShare (string token, string sheetID)

	Delete a share.
bool	ReverseStopSharing (string token, string sheetID) Continue sharing a previously stopped share.
bool	ConvertLiveToStatic (string token, string sheetID) Convert a sheet from static share to live share.
List< Entities.Sheet >	GetLiveSheets (string token)
List< Entities.Sheet >	GetStaticSheets (string token)

PythagoreService.IConnection Interface Reference

Control and monitor the real-time response

Public Member Functions

void	Update () Update connection to extend its timelife
------	--

Properties

string	ID [get, set]
DateTime	LastUpdate [get, set]
ManualResetEventSlim	Lock [get, set]

PythagoreService.INotificationService Interface Reference

Interfaces for notification service

Public Member Functions

RoomResponse	SubscribeRoom (string token, string roomID, long lastUpdate) Register a connection for real-time update in room screen
HomeResponse	SubscribeHome (string token, long lastUpdate) Register a connection for real-time update in home screen

Silverlight Client

Class diagram



Figure 19 - Silverlight client class diagram

Class diagram explanation

Silverlight Client includes 5 screens where end-users can easily interact with.

Sequence Diagrams

UML sequence diagrams are used to show how objects interact in a given situation. The following section will only list some important sequences in some complicated use-cases. This section will show the application flow to accomplish the use-case goals.

1. Drawing use case

- Objective: Helps user create a new figure of their choice on the screen
- Diagram:

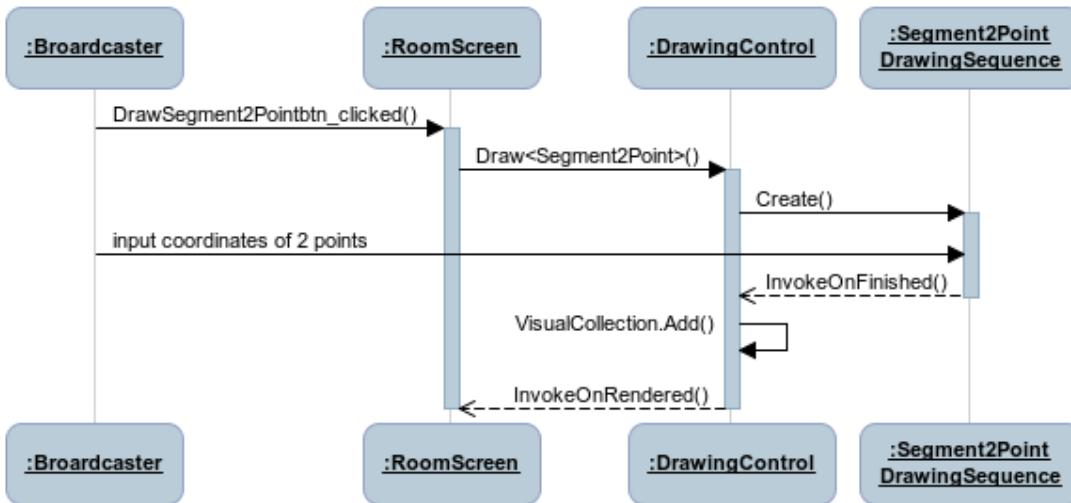


Figure 20 - Drawing figures sequence diagram

- Details description:
 - Broadcaster chooses and clicks a shape/type button on the room screen.
 - The system activates method `Draw<XXX>` from the drawing control, which will create a drawing sequence for the chosen figure.
- After broadcaster inputs necessary input info, the figure will be created and displayed on the room screen.

2. Undo/Redo use case

- Objective: Supports broadcaster to undo/redo actions
- Diagram:

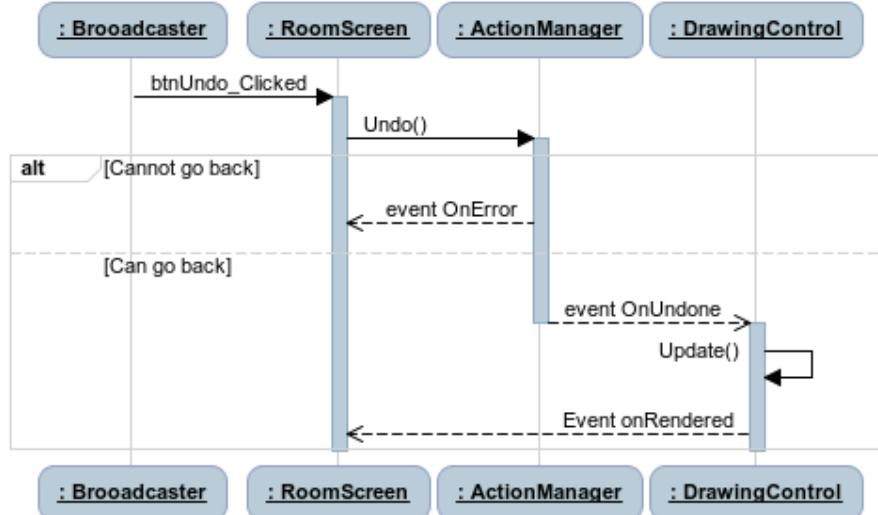


Figure 21 - Undo sequence diagram

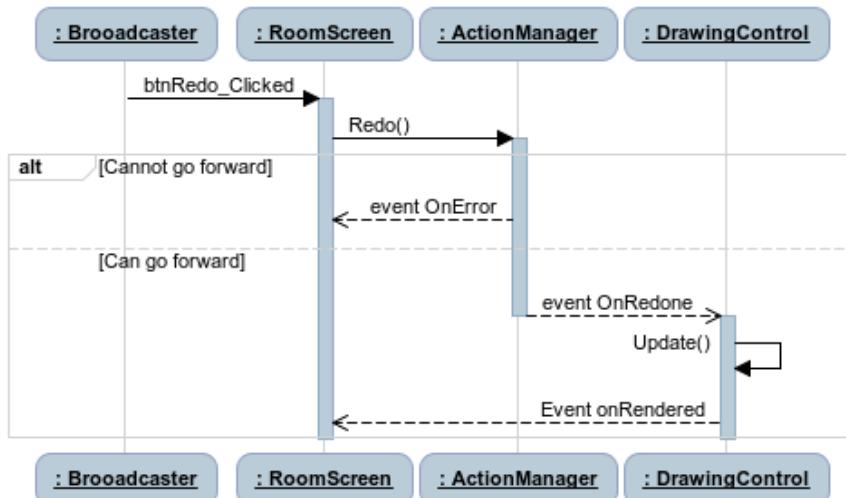


Figure 22 - Redo sequence diagram

- Details description:
 - Broadcaster clicks undo/redo button.
 - System access to the history to get previous/next action.
 - System redraws figures on screen.

3. Save state

- Objective: Supports system to save room state
- Diagram:



Figure 23 - Save state sequence diagram

- Detail description
- Broadcaster performs actions on screen. All actions are recorded by an ActionManager instance. Whenever a change is made, that change is appended to a queue for unsaved commands (namely commandQueue).
- After a period of time, for instance half of a second, client code will test whether it needs to commit changes to server. If any, all of these changes will be removed from commandQueue and added to pendingQueue.
- Commands in pendingQueue will be sent to server to store.
- When server operation completes, pendingQueue is cleared.
- Unless pendingQueue is empty, it's locked for adding.

4. Load State

- Objective: Supports real-time update for client
- Diagram

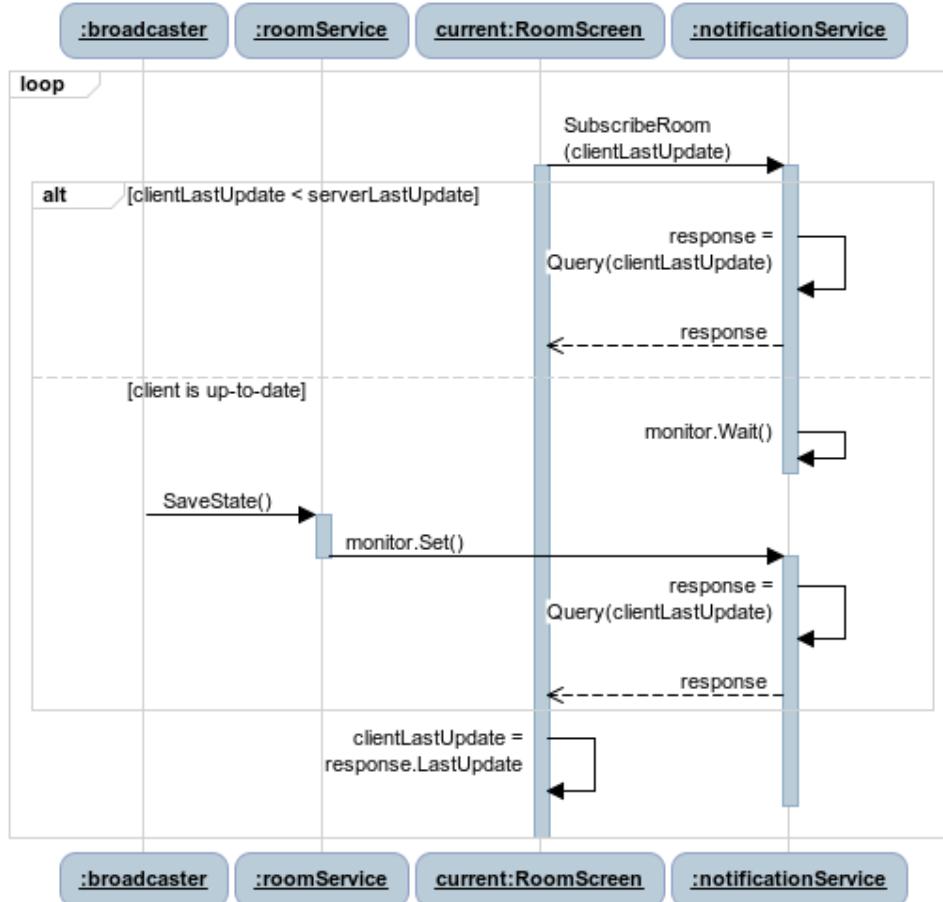


Figure 24 - Load State sequence diagram

- Detail description
 - After initialization, RoomScreen sends a subscription request to server.
 - If there is newer change from other clients, server returns all these changes.
 - Otherwise, server thread will be suppressed by a monitor object.
 - When another thread calls set method from the monitor object, subscription thread awakes and return new changes to client.
 - After receiving changes from server, client continues to subscribe.

5. Export sheet

- Objectives: Supports downloading the sheet as image file or UML file.
- Diagram:

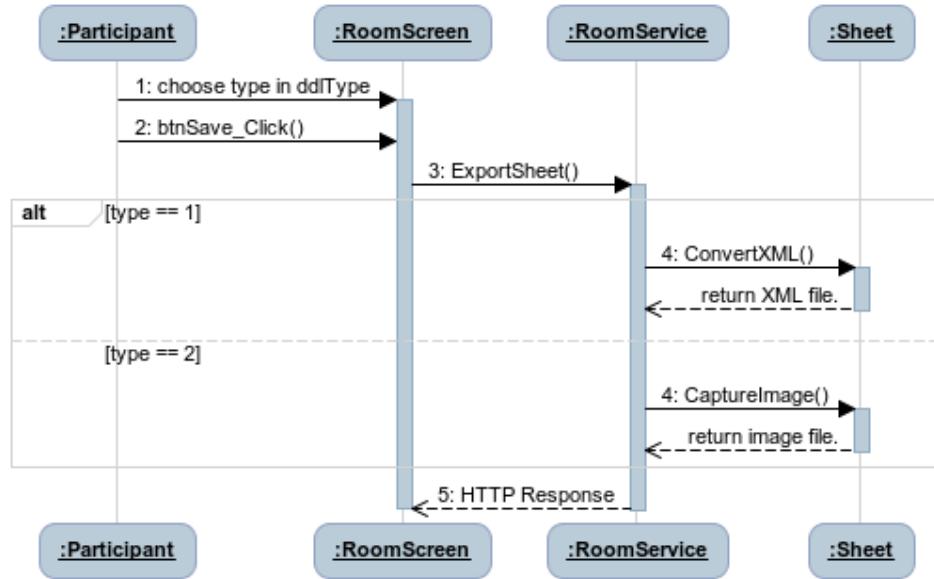


Figure 25 - Export sheet sequence diagram

- Details description:

- 1: Participant selects file type to download
- 2: Participant presses "save" button.
- 3: Activate room service's function to export current sheet
- 4: If file type is XML (1) make current sheet to convert to xml. If file type is image (2) capture the current sheet screen and return image.
- 5: Allow participant to download.

6. Remove room

- Objective: Supports creator to remove a room. Also supports saving shared sheet before deleting a room.
- Diagram:

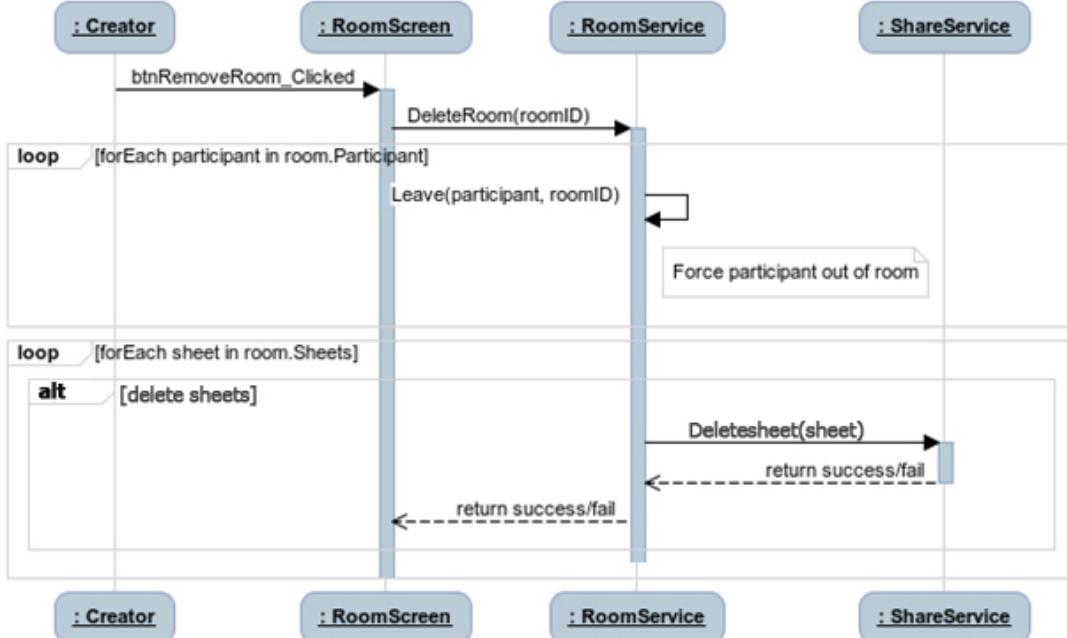


Figure 26 - Remove room sequence diagram

- Details description:
 - Creator chooses room to remove and click button Remove Room.
 - System forces all the Participants out of the room.
 - System deletes all sheets in room.

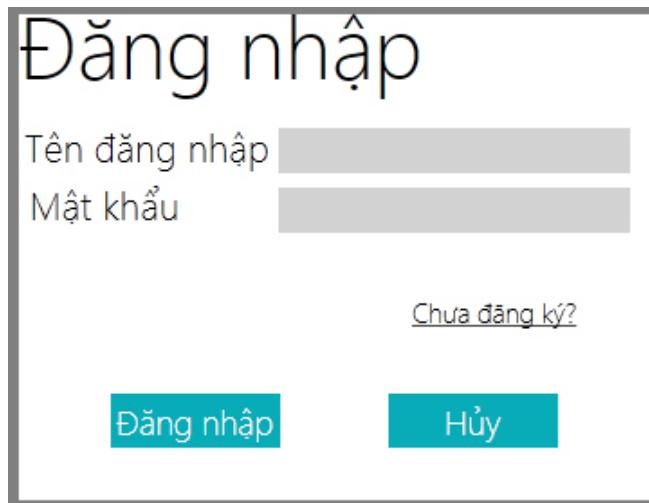
User Interface Design

NOTE:

For more information how to use and interact with graphical elements, see User's Guide (page 136) below.

This section provides detail layout for main screen in Pythagore client. Because Pythagore service is intended to run on Internet Information Services (IIS), it will have no graphical user interface.

Login screen



The login screen features a large title 'Đăng nhập' at the top. Below it are two input fields: 'Tên đăng nhập' and 'Mật khẩu'. A link 'Chưa đăng ký?' is positioned above the buttons. At the bottom are two buttons: 'Đăng nhập' (blue background) and 'Hủy' (white background).

Figure 33 Login screen

Object name	Actions
txtUserName	User types the user name to login
txtPassword	User types the password to login
btnSignIn	User clicks the button to login to website
btnCancel	User clicks the button to cancel
btnOK	Only available in error message box

Register screen



The register screen features a large title 'Đăng ký'. Below it are four input fields: 'Tên đăng nhập', 'Mật khẩu', 'Nhập lại mật khẩu', and 'Địa chỉ mail'. A link 'Đã đăng ký? Chọn vào đây để đăng nhập!' is located below the fields. At the bottom are two buttons: 'Đăng ký' (blue background) and 'Hủy' (white background).

Figure 27 - Register screen

Object name	Actions
txtUserName	User types the user name to register
txtPassword	User types the password to register
txtRetypePassword	User re-types the password to register
txtEmail	User types the email address
txtRegisterLink	User clicks to navigate to Register Page
btnCancel	User clicks the button to cancel
btnRegister	User clicks to register

Home Screen

Room tab

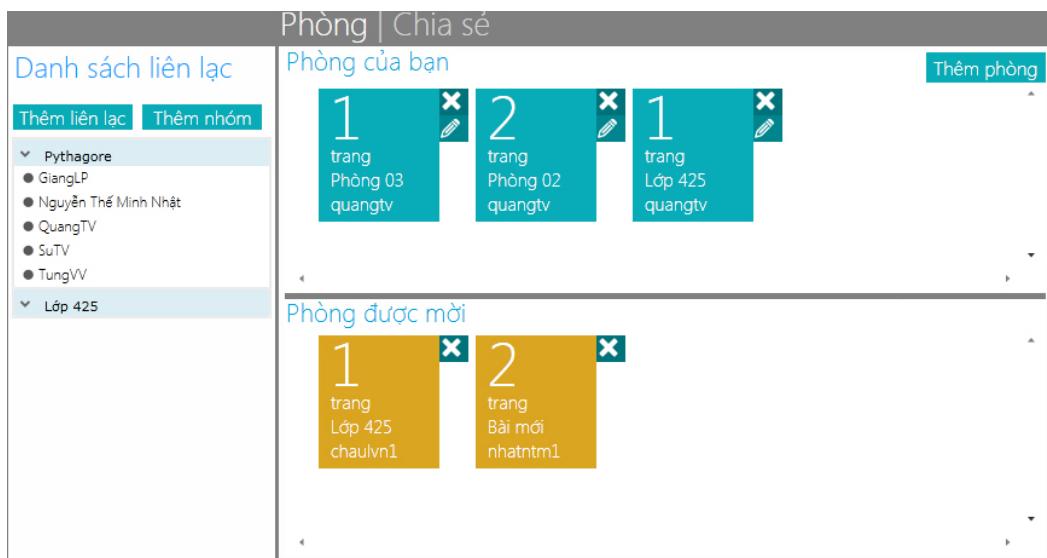


Figure 28 - Home screen: room tab

Object name	Actions
tbRoom	The room tab title
btnAddContact	User uses this button to add a contact to contact list
btnAddGroup	User uses this button to add a group to group list
Contact list item	List of contact already added to contact list
Room list	List of room available
btnAddRoom	User uses this button to add a new room

Share tab



Figure 29 - Home screen: share tab

Object name	Actions
Share list	List of shared sheet that was created by users
tbShare	The Share tab title

Shared sheet clicked

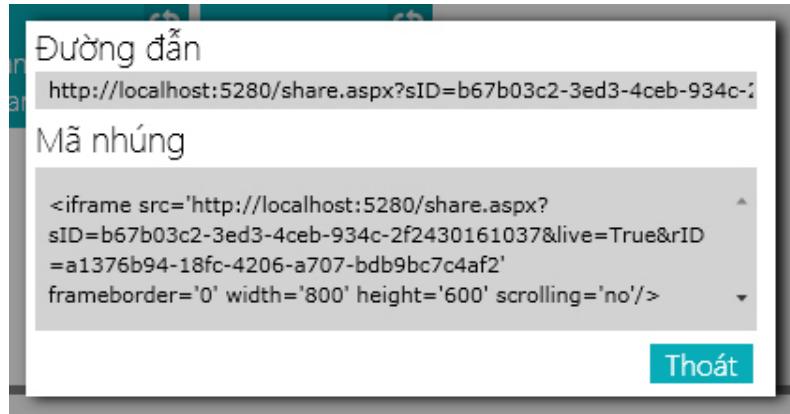


Figure 30 - Share object when clicked

Object name	Actions
txtLink	Textbox contains the link to the sheet
txtEmbedCode	Textbox contains the embedded code
btnCancel	User clicks the button to cancel the action

Add contact, Add group



Figure 31 - Add contact panel and group panel

Object name	Actions
txtContactName	Textbox for user to input contact name
btnAddContact	User clicks the button to add contact to list
btnCancel	User clicks the button to cancel the action
txtGroupName	Textbox for user to input Group name
btnAddgroup	User clicks the button to add the group to list

Edit information

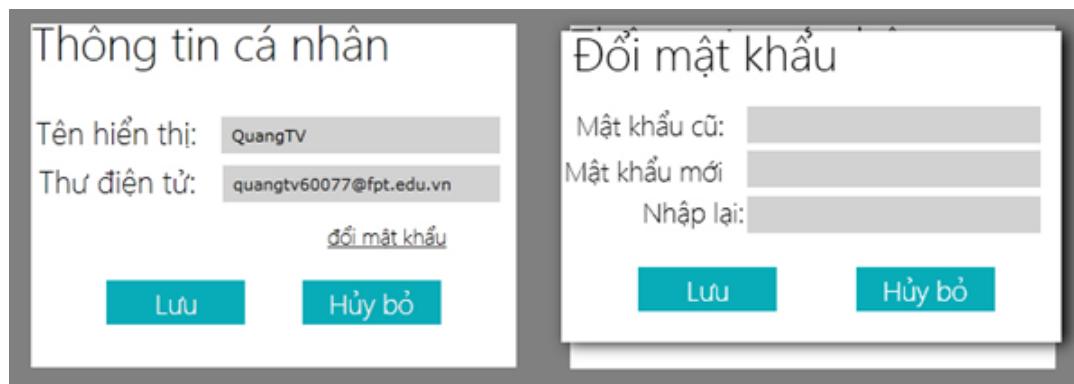


Figure 32 - Edit information screen

Object name	Actions
txtDisplayName	Textbox for user to enter new Display name
txtEmail	Textbox for User to enter new email
btnChangePassword	Button for user to enter change password page
txtOldPassword	Textbox for user to enter old password
txtNewPassword	Textbox for user to enter new password
txtRetype	Textbox for user to retype the new password
btnSave	Save the changed password
btnCancel	Cancel the action

Room screen

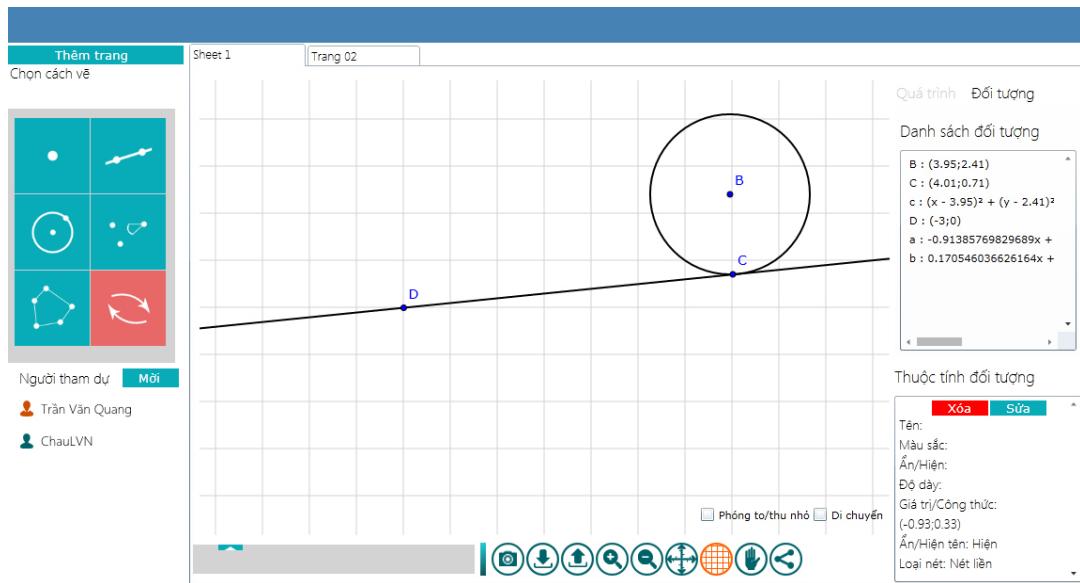


Figure 33 - Room screen

Object name	Actions
sheetItemRoom	Tab Container which contains different tab, each tab is a different lesson
txtChat	Text box for user to enter the message
btnSend	User clicks the button to send the message in text box
btnInvite	User clicks the button to invite new participant
btnAddSheet	User clicks to add new sheet in room
Participant List	List contains the users who are in the room
Object list	List contains the objects on the drawing pad
History List	List contains the history of actions by users
Chatbox	Box contains the message, can be minimize
Properties box	Box contains the properties of the selected object
Drawing toolbox	Contains all the tool to draw objects in pad

Database Design

This section provides detail database structure design, including Table, Fields, and Relationship.

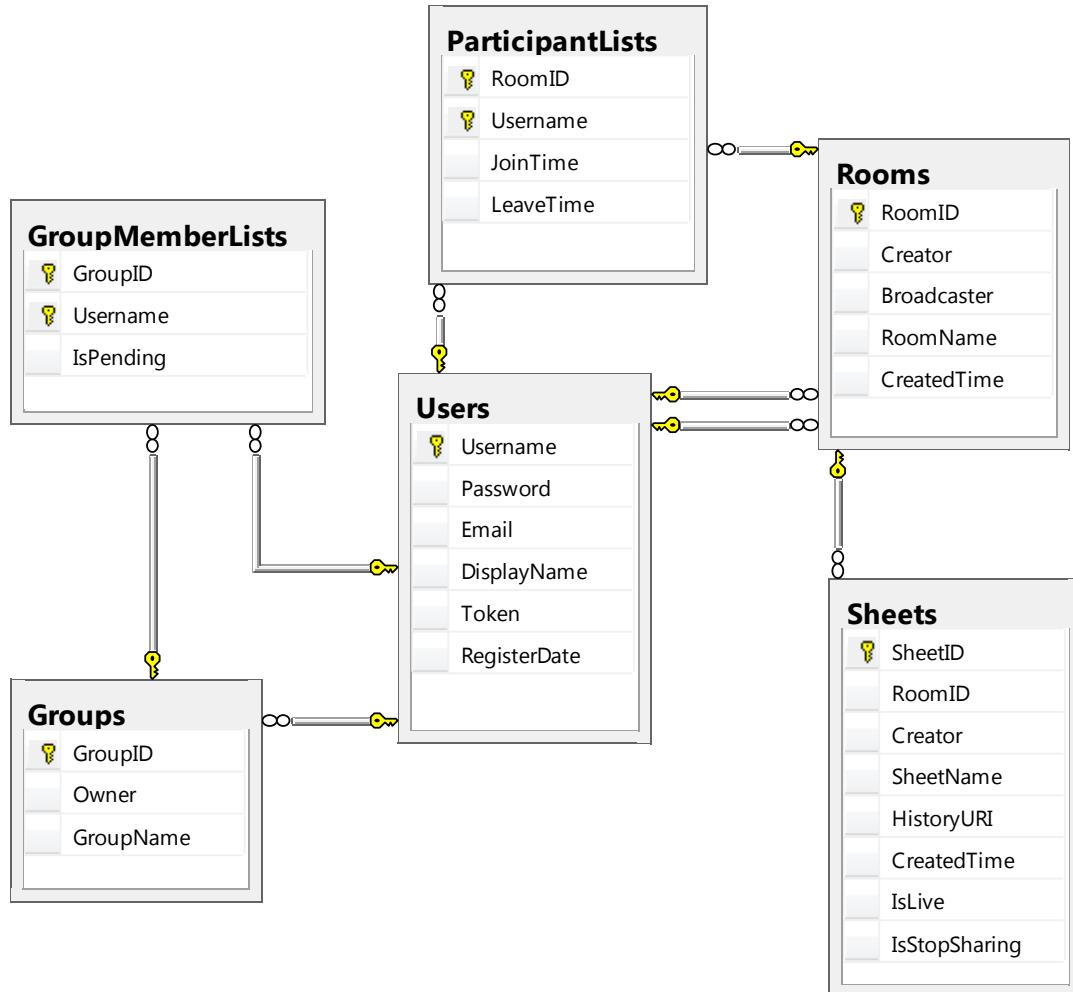


Figure 34 - Pythagore Database Diagram

GroupMemberLists Table

This table is about contact list for each group.

Column Name	Data Type	Allow Null
GroupID	nvarchar(50)	Not allow
Username	nvarchar(50)	Not allow
IsPending	bit	Allow

Groups Table

This table is about group list for each user

Column Name	Data Type	Allow Null
GroupID	nvarchar(50)	Not allow
Owner	nvarchar(50)	Allow

GroupName	nvarchar(255)	Allow
------------------	---------------	-------

ParticipantLists Table

This table is about participant list for each room.

Column Name	Data Type	Allow Null
RoomID	nvarchar(50)	Not allow
Username	nvarchar(50)	Not allow
JoinTime	datetime	Allow
LeaveTime	datetime	Allow

Rooms Table

This table is about room list for each user.

Column Name	Data Type	Allow Null
RoomID	nvarchar(50)	Not allow
Creator	nvarchar(50)	Allow
Broadcaster	nvarchar(50)	Allow
RoomName	nvarchar(100)	Allow
CreatedTime	datetime	Allow

Users Table

This table is about account information list.

Column Name	Data Type	Allow Null
Username	nvarchar(50)	Not allow
Password	nvarchar(100)	Not allow
Email	varchar(100)	Not allow
DisplayName	nvarchar(100)	Not allow
Token	nvarchar(100)	Allow
RegisterDate	datetime	Not allow

Sheets Table

This table is about sheet list for each room.

Column Name	Data Type	Allow Null
SheetID	nvarchar(50)	Not allow
RoomID	nvarchar(50)	Allow
Creator	nvarchar(50)	Allow
SheetName	nvarchar(255)	Allow
HistoryURI	nvarchar(100)	Allow
CreatedTime	datetime	Allow
IsLive	Bit	Not allow
IsStopSharing	Bit	Not allow

Software Test Documentation

Introduction

System overview

This document provides a high-level description of all the tests that have to be performed in the project. It identifies all the features, test plan, test case, checklist of validation, test result, test environment, pass/fail criteria and risk assessment.

Test Approach

- Test Goal: Find out the remaining bugs from the system before releasing to user.
- Test Type: Black box test.
- Test Size: Component Test.
- Test Technique: Checklist.

Test Plan

The test plan helps confirm the quality of the product through testing and validation. The followed parts will describe which feature will or will not be tested, testing tools and test environment.

Features to be tested

There are eight features that will be tested in this document, which are:

Account

- Register
- Login
- Logout
- Edit profile

Contact

- Add Contact
- Decide Contact
- Remove Contact
- Add Group

Room

- Create Room
- Remove Room
- Invite/Accept User
- Remove User
- Leave Room
- Grant Permission
- Take permission
- Raise Hand

Draw

- Angle
- Line
- Curve
- Point
- Polygon
- Transform
- Add text
- Change Properties
- Move
- Remove

Sheet	Share
Add Sheet	Live Embed
Delete Sheet	Static Embed
Rename Sheet	Convert Embed
Export Sheet	Delete Share
Import Sheet	Stop Share
	Revert Share
Chat	Other
Message Transfer	Zoom
Prefer to object	Move

Features not to be tested

N/A

Testing tool and Environment

Test tools:

No special test tool.

Test Environment:

- Computer: Pentium IV 2.00 GHz or above. RAM at least 1GB. HDD free at least 1GB.
- Operating System: Window 7
- Browser: Google Chrome 4+, Firefox 3.5+, Microsoft Internet Explorer 8+
- Platform: Silverlight 5 Runtime

Test Cases

Account Features

1. Case-Register

Purpose

Guest can sign up an account into system database to log in later.

Inputs

- Tên đăng nhập: Username
- Mật khẩu: Password
- Nhập lại mật khẩu: Retype password
- Thư điện tử: Email

Results & Pass/Fail criteria

- Pass: Message "Đăng ký thành công" appears.
- Fail:

- Notification “Xin điền đầy đủ thông tin trước khi bấm đăng nhập” appears if all the informations are not filled.
- Notification “Tên đăng nhập phải từ 8-32 ký tự, có thể dùng "_" và không có ký tự đặc biệt” appear when Username is less than 8 characters or more than 32 characters or have special character.
- Notification “Mật khẩu phải từ 8-32 ký tự” appears when Password is less than 8 characters or more than 32 characters.
- Notification “Mật khẩu và mật khẩu nhập lại không trùng nhau” appears if password and retype password is not the same.
- Notification “Thư điện tử sai cấu trúc” appears when email is in wrong format.
- Notification “Tên đăng nhập đã được sử dụng. Xin thử lại!” appears if username is already used.

TestProcedure

1. Input Username
2. Input Password
3. Input Retype Password
4. Input Email
5. Press “Đăng ký”

2. Case-Login

Purpose

Check if Login Feature is working properly. Registered guest can login into the system with their username and password.

Inputs

- Tên đăng nhập: Username
- Mật khẩu: Password

Results & Pass/Fail criteria

- Pass: Navigate to the Home Screen
- Fail: Show message “Đăng nhập thất bại” or “Tên đăng nhập hoặc mật khẩu không đúng”

Test Procedure

1. Input Username
2. Input Password
3. Press “Đăng nhập”

3. Case-Logout

Purpose

Help user successfully logout of the system

Results & Pass/Fail criteria

- Pass: Log out successfully, return to the Login Page

Test Procedure

From main screen after successfully Login, click "Thoát" link

4. Case-Edit Profile

Purpose

Help user successfully edit theirs profile information

Inputs

From the Main Page, click "Thông tin cá nhân":

- Tên hiển thị: Display name
- Thư điện tử: Email

Thông tin cá nhân

Tên hiển thị: [Placeholder]

Thư điện tử: [Placeholder]

[đổi mật khẩu](#)

Lưu Hủy bỏ

Results & Pass/Fail criteria

- Pass: Show message box: "Cập nhật thành công"
- Fail: Show message "Định dạng thư điện tử sai" or "Tên hiển thị không hợp lệ"

Test Procedure

1. Input new username
2. Input new email
3. Click "Lưu"

5. Case-Change password

Purpose

Help user successfully changing theirs password

Inputs

From the Edit Profile Page, click "Đổi mật khẩu" link.

- Mật khẩu cũ: Old password
- Mật khẩu mới: New password
- Nhập lại: Retype new password

Đổi mật khẩu

Mật khẩu cũ: [Placeholder]

Mật khẩu mới: [Placeholder]

Nhập lại: [Placeholder]

Lưu Hủy bỏ

Results & Pass/Fail criteria

- Pass: Show message box: "Cập nhật mật khẩu thành công"
- Fail: Show message "Định dạng mật khẩu mới sai" or "Cập nhật mật khẩu thất bại"

Test Procedure

1. Input old password
2. Input new password
3. Input retype password
4. Click "Lưu"

Chat Features

1. Case-Message transfer

Purpose

Successfully transfers message between clients when they use Chat feature

Inputs

- From User1 sheet, type in the Chat box: "Hello world"

Results & Pass/Fail criteria

- Pass: From User 2 sheet, Chat block: "User1: Hello world"
- Fail: The message is not transferred

Test Procedure

1. Type: "Hello world"
2. Enter or click button "Gửi"



2. Case-Refer object

Purpose

Checking if the object referred in chat or comment got focused on when user hovers the mouse pointer on its name.

Results & Pass/Fail criteria

- Pass: The referred object gets focused on
- Fail: The referred object does not get focused on

Test Procedure

Hovers the mouse pointer on the object name in the text in chat or comment in a room

Room management Features

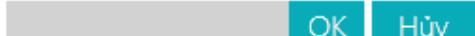
1. Case-Create room

Purpose

Check if a room is successfully created

Inputs

In the room name textbox, input: "Pythagore"



OK

Hủy

Results & Pass/Fail criteria

- Pass: A new sheet with the name "Pythagore" appears on "Phòng của bạn" block
- Fail: No room is created

Test Procedure

1. On the Home Screen, click on the "Thêm phòng" button
2. Enter room name
3. Click "Thêm" button

2. Case-Remove room

Purpose

Check if a room is successfully removed when user deletes it

Results & Pass/Fail criteria

- Pass: When user confirms that they really want to delete the room, the selected room is deleted

- Fail: The room is not removed

Test Procedure

1. On the Home Screen, click on the "X" on the up-right side of the room icon that user wants to remove
2. A confirm message is showed, click OK

3. Case-Invite user

Purpose

Check if an invite request for a user in a room is successfully sent

Inputs

- Tên người được mời: Invited username

Results & Pass/Fail criteria

- Pass: A notification is showing in the room "Đã gửi lời mời" and the receiver receives an invitation to join the room
- Fail: No request have been sent to the receiver

Mời người tham dự

Tên người được mời



Test Procedure

- In the Room Screen, in the "Người tham dự" list, click "Mời" button
- Input target username in "Tên người được mời"
- Click "+" button

4. Case-Accept invite request

Purpose

Test if user is successfully accepting another invite request

Results & Pass/Fail criteria

- Pass: A new orange room appears in user 1's "Phòng được mời" list. A notification is showing on the room "user 1 đã vào phòng" when they enter the invited room
- Fail: User accepts the request but cannot see new room, or user cannot join the invited room.

Test Procedure

1. When message box appear with "Bạn đã được mời vào một phòng mới", click OK
2. Click on the new appeared room (color orange) in "Phòng được mời" list

5. Case-Remove user

Purpose

Test if a user is successfully removed from a room.

Inputs

N/A

Results & Pass/Fail criteria

- Pass: Notification in room with message "User 1 đã bị đuổi khỏi phòng", their name is removed from the "Người tham dự" list. In that user' screen show a message that they are kick out of the room and they cannot rejoin that room unless invited again.
- Fail: User cannot be removed

Test Procedure

1. Click the "X" icon after the username of the participant in "Người tham dự" list
2. Click OK

6. Case-Leave room

Purpose

Test if user is successfully leaving a room

Inputs

N/A

Results & Pass/Fail criteria

- Pass:
 - o If the creator left the room, all the participants of that room will be kicked out, all live-shares are converted into static, and the room is deleted.
 - o If participant left the room, a notification in room with message "User 1 đã rời khỏi phòng" appears, participant's name is deleted from the "Người tham dự" list and cannot rejoin the room unless being re-invited
- Fail: User cannot leaves the room

Test Procedure

1. Click the delete button the room in the "Phòng được mời" list
2. Confirm by clicking "Ok"

7. Case-Grand Permission

Purpose

Test if Broadcaster is successfully giving the drawing permission to another participant

Results & Pass/Fail criteria

- Pass: Notification shows "Đã chuyển quyền vẽ cho thành viên"
- Fail: The broadcaster's right cannot be transferred to another participant

Test Procedure

1. Broadcaster clicks on "Nhường quyền vẽ" icon next to a participant name in "Người tham dự" list
2. Click OK

8. Case-Take Permission

Purpose

Test if Creator is successfully taking the drawing permission back

Inputs

N/A

Results & Pass/Fail criteria

- Pass: Notification shows "Quyền vẽ đã trả về cho chủ phòng"
- Fail: The creator cannot take back the broadcast's right

Test Procedure

Broadcaster clicks on "Đòi quyền vẽ" icon next to the creator name in "Người tham dự" list.

9. Case-Raise hand

Purpose

Test if participant is successfully asking for permission to draw from the creator

Inputs

N/A

Results & Pass/Fail criteria

- Pass: An icon of a hand appears next to the "Người tham dự" list
- Fail: No icon of a hand appears next to the "Người tham dự" list

Test Procedure

Participant clicks on the icon "Giơ tay"

Sheet management Features

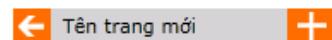
1. Case-Add Sheet

Purpose

Creator can create new sheet at the end, allowing them to use draw and embed function.

Inputs

- Tên trang mới: Type new sheet name here.



Results & Pass/Fail criteria

- Pass: A new sheet with the input name appears in room screen.
- Fail: Notification appears: "Có lỗi khi tạo trang mới"

Test Procedure

1. Log in.
 - 1.1. As room creator
 - 1.2. As room participant
2. Join room
3. Press "Thêm trang" button.
4. Input new sheet name.
5. Press "+" button.

2. Case-Delete Sheet

Purpose

Creator can delete a sheet that they own.

Inputs

N/A

Results & Pass/Fail criteria

- Pass: A sheet is deleted. Notification "User 1 đã xóa trang Sheet1"

- Fail: Sheet cannot be deleted.

Test Procedure

1. Log in.
 - 1.1. As room creator
 - 1.2. As room participant
2. Join room.
3. Add sheet.
4. Press "x" button beside sheet name.
5. Choose "Có" when dialog appears.

3. Case-Rename Sheet

Purpose

Creator can change the name of the sheet as they want.

Inputs

Type new name for sheet in the text box

Sheet 1	✓	✗	trang 2				
---------	---	---	---------	--	--	--	--

Results & Pass/Fail criteria

- Pass: Current sheet name change to new one.
- Fail: Sheet name does not change

Test Procedure

1. Log in.
 - 1.1. As room creator
 - 1.2. As room participant
2. Join room.
3. Press the pen icon next to the sheet name.
4. Input new name.
5. Press check icon.

4. Case-Export Sheet

Purpose

Participant can download the xml for using later or capture the current sheet into image file.

Inputs

N/A

Results & Pass/Fail criteria

- Pass: message box appears "Đã tải về xong". A file with the name "abc.xml" or abc.jpg is downloaded to computer.
- Fail: message box appears: "Có lỗi trong quá trình tải về".

Test Procedure

1. Log in.
2. Join room.
3. Choose export type

4. Press "Tải về máy" button.
5. Press "Chụp màn hình" button.

5. Case-Import Sheet

Purpose

Creator can re-create the sheet using the history of the xml file they have downloaded.

Inputs

- Tên tập tin: File name

Results & Pass/Fail criteria

- Pass: A new blank sheet is created. Message box appears "Đã tải lên xong"
- Fail: Dialog appears: "Không thể kết nối đến máy chủ" or "Tải lên thất bại".

Test Procedure

1. Log in
 - 1.1. As room creator
 - 1.2. As room participant
2. Join room.
3. Press "Tải trang lên" button.
4. Choose file.
5. Press "OK" button.

Contact management Feature

1. Case-Add Contact

Purpose

Users can add a contact to their contact list.

Inputs

- Tên liên lạc: Contact name.
- Nhóm liên lạc: Contact group.

Results & Pass/Fail criteria

- Pass: Notification appears "Đã gửi lời mời". Trigger "Contact decision" when added user logged in.
- Fail: When added user logged in, no message appears.

Test Procedure

1. Log in.
2. Press "Thêm liên lạc" button.
3. Input contact name.
4. Select group to put contact.
5. Press "Thêm" button.

2. Case-Contact Decision

Purpose

Users can accept or deny a request.

Results & Pass/Fail criteria

- Pass: Adding user appear in contact list. If user does not have a group, it will be automatically created with name "Nhóm tự động tạo"
- Fail: No user appear in contact list.

Test Procedure

1. Log in as adding user.
2. Perform add contact.
3. Log out.
4. Log in as added user.
5. Decision
 - 5.1. Press "Chấp nhận" button.
 - 5.2. Press "Từ chối" button.

3. Case-Remove Contact

Purpose

Users can remove a contact from their contact list

Results & Pass/Fail criteria

- Pass: User is removed from contact list.
- Fail: Nothing happen.

Test Procedure

1. Log in.
2. Click at the "X" icon that appears next to the contact.
3. Choose "Ok".

4. Case-Add Group

Purpose

Users can add new group.

Inputs

- Tên nhóm: Group's name.

Results & Pass/Fail criteria

- Pass: A group is created with the input name.
- Fail: No group is created.

Test Procedure

1. Log in
2. Press "Thêm nhóm" button.
3. Input group name.
4. Press "Thêm" button.

Thêm nhóm

Tên nhóm

Thêm

Hủy

Share Features

1. Case-Static Embed

Purpose

Support users to share the static content of the sheet to be embedded to other website.

Inputs

N/A

Results & Pass/Fail criteria

- Pass: A new static share is appeared on share tab. Notification "Đã tạo trang chia sẻ tĩnh thành công"
- Fail: No share is created.

Test Procedure

1. Log in.
 - 1.1. As room creator
 - 1.2. As room participant
2. Join Room.
3. Click the "Chia sẻ trang" icon
4. Choose "Chia sẻ tĩnh".

2. Case-Lived Embed

Purpose

Support users to share a sheet to other websites.

Results & Pass/Fail criteria

- Pass: The live content of the sheet appear on share tab. Notification "Đã tạo trang chia sẻ động thành công"
- Fail: No share is created.

Test Procedure

1. Log in.
 - 1.1 As room creator
 - 1.2 As room participant
2. Join room.
3. Press "Chia sẻ trang" icon.
4. Choose "Chia sẻ động"

3. Case-Convert Embed

Purpose

Convert a sheet from live embedded to static embedded.

Results & Pass/Fail criteria

- Pass: A textbox appears with message: "Đã chuyển dạng chia sẻ", the live share become static share.
- Fail: No share is converted.

Test Procedure

1. Log In.
2. Choose "Chia sẻ" Tab.
3. Press "Chuyển sang chia sẻ tĩnh" icon next to the room.

4. Case-Delete Share

Purpose

Creator can delete desired share after confirming.

Results & Pass/Fail criteria

- Pass: The selected share is deleted. A message box is showed "Đã xóa chia sẻ"
- Fail: The selected share is not deleted.

Test Procedure

1. Log In.
2. Choose "Chia sẻ" Tab.
3. Press "Xóa" icon next to the room.
4. Press "OK" button.

5. Case-Stop Share

Purpose

Creator successfully stops a share after confirming.

Results & Pass/Fail criteria

- Pass: The share is no longer appears in other webs that embed this share. Message box "Đã dừng chia sẻ" is showed.
- Fail: The selected share is not stop-sharing.

Test Procedure

1. Log In.
2. Choose "Chia sẻ" Tab.
3. Press "Ngừng chia sẻ" button.

6. Case-Revert Share

Purpose

Support users to continue sharing a sheet that previously stopped

Results & Pass/Fail criteria

- Pass: The share is now able to appear in other webs that embed this share. Message box "Đã tiếp tục chia sẻ" is showed.
- Fail: The selected share cannot resume sharing.

Test Procedure

1. Log In.
2. Choose "Chia sẻ" Tab.
3. Press "Ngừng chia sẻ" button.
4. Press "Tiếp tục chia sẻ" button.

Draw Features

1. Case-Draw Angle Geometry

Purpose

User can draw and create angle and all of its variants into sheet.

Inputs

- Điểm: Point – Draw by click on blank area or select by click on any exist point.
- Đường thẳng: Line – Select by click on any exist line
- Góc: Angle – Input by user

Results & Pass/Fail criteria

- Pass: A success message is displayed. And the desired figure is drawn.
- Fail: No figures drawn.

Test Procedure

1. Login
2. Join room
3. Draw figures
 - 3.1. Press "Góc giữa hai đường thẳng" button
 - Choose first linear
 - Choose second linear
 - 3.2. Press "Góc với số đo cho trước" button
 - Choose first point
 - Choose second point
 - Input angle
 - 3.3. Press "Góc giữa ba điểm" button
 - Choose first point
 - Choose second point
 - Choose third point

2. Case-Draw Point Geometry

Purpose

User can draw and create point and all of its variants into sheet.

Inputs

- Điểm: Point – Draw by click on blank area or select by click on any exist point.
- Đường thẳng: Line – Select by click on any exist line.
- Đường cong: Curve – Select by click on any exist curve.

Results & Pass/Fail criteria

- Pass: A success message is displayed. And the desired figure is drawn.
- Fail: No figures drawn.

Test Procedure

1. Log in

- 1.1. As room broadcaster
- 1.2. As room participant
2. Join room
3. Draw Figures
 - 3.1. Press "Vẽ một điểm" button.
 - Draw point.
 - 3.2. Press "Vẽ trung điểm" button.
 - Draw point #1
 - Draw point #2
 - 3.3. Press "Vẽ giao điểm" button
 - Select Line #1
 - Select Line #2
 - 3.4. Press "Vẽ giao điểm" button
 - Select Line
 - Select Curve
 - 3.5. Press "Vẽ giao điểm" button
 - Select Curve
 - Select Line
 - 3.6. Press "Vẽ giao điểm" button
 - Select Curve #1
 - Select Curve #2

3. Case-Draw Line Geometry

Purpose

User can draw and create line and all of its variants into sheet.

Inputs

- Điểm: Point – Draw by click on blank area or select by click on any exist point.
- Đường thẳng: Line – Select by click on any exist line.
- Đường cong: Curve – Select by click on any exist curve.

Results & Pass/Fail criteria

- Pass: A success message is displayed. And the desired figure is drawn.
- Fail: No figures drawn.

Test Procedure

1. Log in
 - 1.1. As room broadcaster
 - 1.2. As room participant
2. Join room
3. Draw Figures
 - 3.1. Press "Vẽ đoạn thẳng" button.
 - Draw point #1.

- Draw point #2.
- 3.2. Press “Vẽ đường thẳng” button
- Draw point #1.
 - Draw point #2.
- 3.3. Press “Vẽ véc-tơ” button
- Draw point #1.
 - Draw point #2.
- 3.4. Press “Vẽ tia” button
- Draw point #1.
 - Draw point #2.
- 3.5. Press “Vẽ đường thẳng song song” button
- Select line
 - Draw point
- 3.6. Press “Vẽ đường thẳng vuông góc” button
- Select line
 - Draw point
- 3.7. Press “Vẽ tiếp tuyến đường cong” button
- Select curve
 - Draw point
- 3.8. Press “Vẽ đường trung trực” button
- Select point #1
 - Select point #2
- 3.9. Press “Vẽ đường trung trực” button
- Select line
- 3.10. Press “Vẽ đường phân giác” button
- Select point #1.
 - Select point #2.
 - Select point #3.
- 3.11. Press “Vẽ đường phân giác” button
- Select line #1.
 - Select line #2.

4. Case-Draw Curve Geometry

Purpose

User can draws and creates curve and all of its variants into sheet.

Inputs

- Point – Draw by click on blank area or select by click on any exist point.
- Line – Select by click on any exist line.

Results & Pass/Fail criteria

- Pass: A success message is displayed. And the desired figure is drawn.
- Fail: No figures drawn.

Test Procedure

1. Log in
 - 1.1. As room broadcaster
 - 1.2. As room participant
2. Join room
3. Draw Figures
 - 3.1. Press "Vẽ cung bán hình tròn" button.
 - Draw point #1.
 - Draw point #2
 - 3.2. Press "Vẽ cung thông qua 3 điểm" button.
 - Draw point #1.
 - Draw point #2.
 - Draw point #3.
 - 3.3. Press "Vẽ cung đi qua 3 điểm" button
 - Draw point #1.
 - Draw point #2.
 - Draw point #3.
 - 3.4. Press "Vẽ hình tròn có tâm và bán kính" button
 - Draw point.
 - Input number.
 - Press "OK" button.
 - 3.5. Press "Vẽ hình tròn có tâm và qua 1 điểm" button
 - Draw point #1.
 - Draw point #2.
 - 3.6. Press "Vẽ hình tròn đi qua 3 điểm" button
 - Draw point #1.
 - Draw point #2.
 - Draw point #3.
 - 3.7. Press "Vẽ hình ellipse đi qua 3 điểm" button
 - Draw point #1.
 - Draw point #2.
 - Draw point #3.
 - 3.8. Press "Vẽ hình pa-ra-bôn" button
 - Draw point.
 - Select line.
 - 3.9. Press "Vẽ hình hi-péc-bôn" button
 - Draw point #1.
 - Draw point #2.
 - Draw point #3.

5. Case-Draw Polygon Geometry

Purpose

User can draws and creates polygon and all of its variants into sheet.

Inputs

- Điểm: Point – Draw by click on blank area or select by click on any exist point.

Results & Pass/Fail criteria

- Pass: A success message is displayed. And the desired geometry is drawn.
- Fail: No figures drawn.

Test Procedure

1. Log in
 - 1.1. As room broadcaster
 - 1.2. As room participant
2. Join room
3. Draw Figures
 - 3.1. Press "Vẽ đa giác" button.
 - Draw point #1.
 - Draw point #2.
 - Draw point #3.
 - Draw point #4.
 - Draw point #5.
 - Draw point #6.
 - Draw point #7.
 - Draw point #8.
 - Draw point #9.
 - Select point #1.
 - 3.2. Press "Vẽ đa giác đều" button.
 - Draw point #1.
 - Draw point #2.
 - Input number

6. Case-Draw Tangent

Purpose

Check if user can create a tangent of a conic by choosing a conic and a point.

Inputs

- Input from mouse click: choose a conic then a point

Results & Pass/Fail criteria

- Pass: If point is on conic, only one tangent is created. Else two is created.
- Fail: No tangent is created.

Test Procedure

1. Choose tangent icon
2. Choose a conic

3. Choose a point

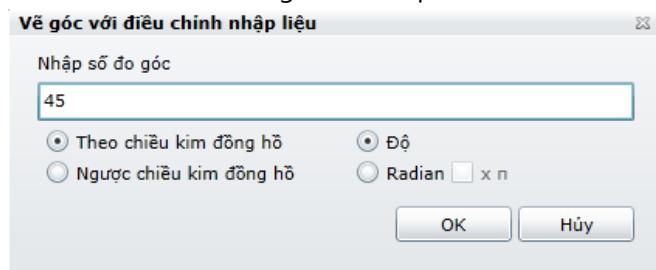
7. Case-Draw-Rotation

Purpose

Check if user can create a new figure by rotating a chosen figure with a chosen angle around a selected center

Inputs

- Nhập số đo góc: Rotating angle
- Input from mouse click: Choose a figure and a point as center



Results & Pass/Fail criteria

- Pass: A rotated of the chosen figure was created according to the input
- Fail: The angle cannot be created; the angle created has wrong value or wrong direction

Test Procedure

1. Input the rotating angle
2. Choose a figure
3. Choose a point

8. Case- Draw-Translation

Purpose

Check if user can create a new figure by translating a chosen figure by a vector

Inputs

- Input from mouse click: Choose a figure and a vector

Results & Pass/Fail criteria

- Pass: a translated figure was created according to the vector
- Fail: The angle cannot be created; the angle created has wrong value

Test Procedure

1. Choose a figure
2. Choose a vector

9. Case- Draw-Reflection

Purpose

Check if user can create a new figure by reflecting a chosen figure by a point or a linear object

Inputs

- Input from mouse click: Choose a figure and a point or a linear object

Results & Pass/Fail criteria

- Pass: a reflected figure was created according to the input
- Fail: The angle cannot be created; the angle created has wrong value

Test Procedure

1. Choose a figure
2. Choose a point or a linear object

10. Case-Change properties

Purpose

Test if user can change object property in the room screen

Inputs

- Tên: Object's name
- Màu sắc: Color
- Ẩn/Hiện: Hidden/Show figure
- Độ dày: Thickness
- Ẩn/Hiện tên: Hidden/Show name
- Loại nét: Line style

Tên:	c
Màu sắc:	[Color Swatch]
Ẩn/Hiện:	Ẩn
Độ dày:	2
Ẩn/Hiện tên:	<input checked="" type="checkbox"/> Ẩn
Loại nét:	Nét liền
Lưu Hủy	

Results & Pass/Fail criteria

- Pass: Object is changed according to the new property value
- Fail: Object is not changed according to the new property value

Test Procedure

1. Click on object
2. From Property table, modify value
3. Click "Lưu"

11. Case-Move

Purpose

Test if broadcaster can move selected object to another location

Inputs

N/A

Results & Pass/Fail criteria

- Pass: object is successfully moved to another location
- Fail: Object cannot be moved

Test Procedure

1. Select object
2. Drag and drop object to another location

12. Case-Remove

Purpose

Test if broadcaster can remove an object on sheet

Inputs

N/A

Results & Pass/Fail criteria

- Pass: Selected object is successfully removed from sheet
- Fail: object is not removed

Test Procedure

1. Select object
2. Click "Xóa" button in "Thuộc tính đối tượng" table or Del button

13. Case-Undo

Purpose

Test if user can undo an action

Inputs

N/A

Results & Pass/Fail criteria

- Pass: Action is successfully undone
- Fail: Action is not undone, or undone not according to the "Quá trình" list

Test Procedure

In the "Quá trình" table, click button "←" or Ctrl+Z

14. Case-Redo

Purpose

Test if user can redo an action

Inputs

N/A

Results & Pass/Fail criteria

- Pass: Action is successfully redone
- Fail: Action is not redone, or redone not according to the "Quá trình" list

Test Procedure

In the "Quá trình" table, click button "→" or Ctrl+Y

15. Case-Zoom

Purpose

Test if the screen can be zoom in and zoom out

Inputs

N/A

Results & Pass/Fail criteria

- Pass: User can zoom in or zoom out with mouse or icon

Test Procedure

1. Check checkbox "Move" (if using mouse to zoom)
2. Scroll
3. Click on icon "Phóng to" or "Thu nhỏ"

16. Case-Move screen

Purpose

Check if user can change the center of the screen

Results & Pass/Fail criteria

- Pass: user can drag the screen as they want

Test Procedure

1. Check checkbox "Move"

2. Drag the screen

Checklists

Abbreviation:

State: [A]bnormal – [N]ormal – [B]oundary

Result: [P]ass – [F]ail

Feature account

Register

Case no.	Fields	Input	Output message	State	Result
AM001	Tên đăng nhập	Q	Tên đăng nhập phải từ 8-32 ký tự, có thể dùng "_" ở giữa và không có ký tự đặc biệt	A	P
	Mật khẩu	12345678			
	Nhập lại Mật khẩu	12345678			
	Thư điện tử	Quang@fpt.edu.vn			
AM002	Tên đăng nhập	QuangTV1234	Mật khẩu phải từ 8-32 ký tự	A	P
	Mật khẩu	12345#			
	Nhập lại Mật khẩu	12345#			
	Thư điện tử	Quang@fpt.edu.vn			
AM003	Tên đăng nhập	QuangTV1234	Mật khẩu và mật khẩu nhập lại không trùng nhau	A	P
	Mật khẩu	12345678			
	Nhập lại Mật khẩu	12345			
	Thư điện tử	Quang@fpt.edu.vn			
AM004	Tên đăng nhập	QuangTV1234	Thư điện tử sai cấu trúc	A	P
	Mật khẩu	12345678			
	Nhập lại Mật khẩu	12345678			
	Thư điện tử	Quangfpt.edu.vn			
AM005	Tên đăng nhập	QuangTV1	Tên đăng nhập đã được sử dụng. Xin thử lại!	A	P
	Mật khẩu	12345678			
	Nhập lại Mật khẩu	12345678			
	Thư điện tử	Quang@fpt.edu.vn			
AM006	Tên đăng nhập	QuangTV1234	Đăng ký thành công!	N	P

	Mật khẩu	1234_#%		
	Nhập lại Mật khẩu	1234_#%		
	Thư điện tử	Quang@fpt.edu.vn		

Login

Case no.	Fields	Input	Output	State	Result
AM007	Tên đăng nhập	Quangtv123	Message: Tên đăng nhập hoặc mật khẩu không đúng	A	P
	Mật khẩu	12345678			
AM008	Tên đăng nhập	QuangTV1234	Navigate to Home Screen	N	P
	Mật khẩu	12345678			

Logout

Case no.	Input	Output	State	Result
AM009	Click "Thoát"	Navigate to Login page, delete session	N	P

Edit profile

Case no.	Fields	Input	Output message	State	Result
AM01 0	Tên đăng nhập	Quangtv123456	Cập nhật thành công	N	P
	Thư điện tử	Quang@gmail.com			
AM01 1	Tên đăng nhập		Vui lòng nhập đầy đủ thông tin trước khi bấm "Lưu"	A	P
	Thư điện tử	Quang@gmail.com			
AM01 2	Tên đăng nhập	_Quangtv	Tên đăng nhập phải từ 8-32 ký tự, có thể dùng "_" ở giữa và không có ký tự đặc biệt	A	P
	Thư điện tử	Quang@gmail.com			
AM01 3	Tên đăng nhập	Quangtv123456	Thư điện tử sai cấu trúc	A	P
	Thư điện tử	Quang@gmail			
AM01 4	Tên đăng nhập	NhatNTM1	Tên đăng nhập đã được sử dụng. Xin thử lại!	A	P
	Thư điện tử	NhatNTM1@gmail.com			

Change password

Case no.	Fields	Input	Output message	State	Result
----------	--------	-------	----------------	-------	--------

AM015	Mật khẩu cũ	1234	Mật khẩu cũ nhập sai	A	P
	Mật khẩu mới	Quang123456			
	Nhập lại	Quang123456			
AM016	Mật khẩu cũ	12345678	Mật khẩu phải từ 8-32 ký tự	A	P
	Mật khẩu mới	Quang			
	Nhập lại	Quang			
AM017	Mật khẩu cũ	12345678	Mật khẩu và mật khẩu nhập lại không tương đương	A	P
	Mật khẩu mới	Quang123456			
	Nhập lại	Quang12345			
AM018	Mật khẩu cũ	12345678	Cập nhật mật khẩu thành công	N	P
	Mật khẩu mới	Quang123456			
	Nhập lại	Quang123456			

Feature chat

Message transfer

Case no.	Fields	Input	Output message	State	Result
CH001	Chatbox	Hello world	Other participants: "User1: Hello world"	N	P

Refer object

Case no.	Input	Output	State	Result
CH002	Mouse hover on "Đường thẳng a" in comment or chat	Focus on line with the name a	N	P

Room management

Create room

Case no.	Fields	Input	Output	State	Result
RM001	Tên phòng	Phòng 1	New room added in "Phòng của bạn" with name "Phòng 1"	N	P

Remove room

Case no.	Input	Output	State	Result
RM002	Click on the "X" icon of the room	Ask for confirm before delete. Room is removed in "Phòng của bạn"	N	P

Invite user

Case no.	Fields	Input	Output	State	Result
RM003	Tên người được mời	TungVV 1	Notification in screen "Đã gửi lời mời". Notification in Tung's screen "Bạn được mời vào một phòng mới", new room appears in "Phòng được mời"	N	P

Accept invite request

Case no.	Input	Output	State	Result
RM004	Click in the new appeared room that user is invited (which have orange color)	A notification is show on the sheet "Tùng đã vào phòng", invited user appear on the "Người tham dự" list	N	P

Remove user

Case no.	Input	Output	State	Result
RM005	Choose user in the "Người tham dự" list and click "Xóa" button	Selected user is removed from room	N	P

Leave room

Case no.	Input	Output	State	Result
RM006	User delete invited room	A notification is show on the sheet "User A đã rời khỏi phòng", user name is remove from "Người tham dự" list	N	P

Grand permission

Case no.	Input	Output	State	Result
RM008	Broadcaster click "Cho phép vẽ" icon	The participant name on "Người tham dự" list is underline; they are now a broadcaster and can draw.	N	P

Take permission

Case no.	Input	Output	Result	State
RM008	Broadcaster click "Đòi quyền vẽ" icon	A notification appears "Quyền vẽ đã trả về cho chủ phòng". Room creator is now a broadcaster and can draw.	N	P

Raise hand

Case no.	Input	Output	Result	State
RM007	Participant click "Giơ tay" icon	A hand icon is showed next to the participant name on "Người tham dự" list. Notification "Đã giơ tay thành công"	N	P

Sheet management

Add Sheet

Case No.	Input	Output	Result	State
SH001	Case 1.1 New sheet name: "New sheet"	A sheet is created.	N	P
SH002	Case 1.1	Unable to continue step 3.	N	P

Delete Sheet

Case No.	Input	Output	Result	State
SH003	Case 1.1	A sheet is delete	N	P
SH004	Case 1.2	Unable to continue step 3.	N	P
SH005	Repeat case 1.1 until no sheets left	A new sheet automatically is created	N	P

Rename Sheet

Case No.	Input	Output	Result	State
SH006	Case 1.1	Sheet name is changed.	N	P

SH007	Case 1.2	Unable to continue step 3.	N	P
-------	----------	----------------------------	---	---

Export Sheet

Case No.	Input	Output	State	Result
SH008	Case 3.1	An xml file is downloaded.	N	P
SH009	Case 3.2	A picture is downloaded.	N	P

Import Sheet

Case No.	Input	Output	State	Result
SH010	Case 1.1 File content blank.	A blank sheet is created.	N	P
SH011	Case 1.1 File content: "abcdefghi"	Dialog appears: "Có lỗi khi đọc file upload. Vui lòng refresh lại trang". A blank sheet is created.	A	P
SH012	Case 1.2	Unable to continue step 3.	N	P

Contact management

Add Contact

Case No.	Input	Output	State	Result
CO001	Invited User: "TungVV1"	When TungVV1 log in, trigger Decide Contact.	N	P
CO002	Invited User: "ABCDEF".	Notification appears: "Không thể gửi lời mời".	N	P

Decide Contact

Case No.	Input	Output	State	Result
CO003	Case 5.1 Inviting User: "ChauLVN1" Invited User: "TungVV1"	User ChauLVN1 appear on TungVV1 contact screen.	N	P
CO004	Case 5.2 Inviting User: "ChauLVN1" Invited User: "TungVV1"	Dialog appears: "Người dùng TungVV1 đã từ chối lời đề nghị" when ChauLVN1 log in.	N	P

Remove Contact

Case No.	Input	Output	Result	State
CO005	Removed User: "TungVV1"	Contact TungVV1 disappear on contact list.	N	P

Add Group

Case No.	Input	Output	Result	State
CO006	Group Name: "ABC"	A group with the name "ABC" appears on contact list.	N	P
CO007	Repeat procedure twice. Group Name: "DEF"	Dialog appears: "Tên nhóm đã tồn tại. Xin thử lại".	N	P

Share feature

Live Share

Case No.	Input	Output	Result	State
SA001	Case 1.1.	The new share is appeared on live area of "chia sẻ" tab.	N	P
SA002	Case 1.2	Unable to continue step 3.	N	P

Static Share

Case No.	Input	Output	Result	State
SA003	Case 1.1.	The new share is appeared on static area of "chia sẻ" tab.	N	P
SA004	Case 1.2	Unable to continue step 3.	N	P

Convert Share

Case No.	Input	Output	Result	State
SA007	Share "abc" on live area.	Share "abc" on static area.	N	P

Delete Share

Case No.	Input	Output	State	Result
SA008	Share "abc".	Share "abc" disappear from share tab.	N	P

Stop Share

Case No.	Input	Output	State	Result
SA009	Share "abc".	Share "abc" is no longer show on embedding website.	N	P
SA010	Repeat procedure twice. Share "abc".	Unable to continue step 3 of second time.	N	P

Revert Share

Case No.	Input	Output	State	Result
SA011	Share "abc".	Share "abc" is still not change on embedding website.	N	P
SA012	Repeat step 4 twice. Share "abc".	Unable to continue step 4 for second time.	N	P

Draw feature

Draw Angle

Case no.	Fields	Input	Output	State	Result
DR001	Nhập góc	90	An angle is created from 2 chosen points clockwise with value egal 90 degree		P
	Theo chiều kim đồng hồ	Checked			
	Ngược chiều kim đồng hồ	Uncheck			
	Độ	Checked			
	Radian	Uncheck			
DR002	Nhập góc	90	An angle is created from 2 chosen points counter clockwise with value egal 90 degree		P
	Theo chiều kim đồng hồ	Uncheck			
	Ngược chiều kim đồng hồ	Checked			
	Độ	Checked			

	Radian	Uncheck			
DR003	Nhập góc	aaa	Message box "aaa° không hợp lệ. Cần nhập một số đo góc"	A	P
	Theo chiều kim đồng hồ	Uncheck			
	Ngược chiều kim đồng hồ	Checked			
	Độ	Checked			
	Radian	Uncheck			
DR004	Nhập góc	3.14	An angle is created from 2 chosen points clockwise with value equal 180degree	N	P
	Theo chiều kim đồng hồ	Checked			
	Ngược chiều kim đồng hồ	Uncheck			
	Độ	Uncheck			
	Radian	Checked			
DR005		Choose 3 points	An angle is created from 3 chosen points	N	P
DR006		Choose 2 lines	An angle is created from 2 chosen lines	N	P

Draw Point

Case No.	Field	Input	Output	State	Result
DR007	Case	1.2	Unable to continue step 3	N	P
DR008	Case	3.1	A point with coordinate (1,1)	N	P
	Point	Point (1,1)			
DR009	Case	3.2	Three point with coordinate (1,1); (3,3); (2,2)	N	P
	Point	Point (1,1)			
	Point	Point (3,3)			
DR012	Case	3.3	A new point with coordinate (3,3)	N	P
	Line	Point (1,1) Point (2,2)			
	Line	Point (3,1) Point (3,2)			
DR013	Case	3.4	A new points with coordinate (1.56,1.56)	N	P
	Line	Point (1,1) Point (2,2)			
	Semicircle	Point (-4,0) Point (2,0)			
DR014	Case	3.4	Two new points with coordinate	N	P

	Line	Point (0,2) Point (6,2)	(16.56,2); (7.42,2)		
	Circular Arc	Point (7,0) Point (12,0) Point (0,2)			
DR015	Case	3.4	A new points with coordinate (-7.13,2)	N	P
	Line	Point (1,1) Point (2,2)			
	Circumcircular Arc	Point (0,0) Point (-6,0) Point (0,6)			
DR016	Case	3.5	Two new points with coordinate (3,3); (4,4)	N	P
	Line	Point (1,1) Point (2,2)			
	Circle	Point (7,0) Point (2,0)			
DR017	Case	3.5	Two new points appear with coordinate (2.37,2.37), (0.63, 0.63)	N	P
	Line	Point (1,1) Point (2,2)			
	Parabola	Line (2,0); (3;1) Point (2,1)			
DR018	Case	3.6	Four new points appear: (7.57,-1.97), (3.79,-4.92), (1.03,4.57), (-1.92,0.79)	N	P
	Circle	Point (3,0) Point (8,0)			
	Hyperbola	Point (2,0) Point (3,-1) Point (4,-1)			
DR019	Case	3.6	Two new points appear	N	P
	Ellipse	Point (2,0) Point (4,0) Point (5,0)			
	Circumcircular Arc	Point (1,1) Point (3,0) Point (5,1)			
DR020	Case	3.6	Three new points appear with coordination (2,2), (-2,-0.8), (-2,4.8)	N	P
	Ellipse	Point (-6,2)			
		Point (1,2)			
		Point(2,2)			
	Circle	Point (-1,2)			
		Point(2,2)			

Draw Line

Case No.	Field	Input	Output	State	Result
DR020	Case	1.2	Unable to continue step 3	N	P
DR021	Case	3.1	A segment appear	N	P
	Point	Point (1,1)			
	Point	Point (2,2)			
DR022	Case	3.2	A line appear	N	P
	Point	Point (1,1)			
	Point	Point (2,2)			
DR023	Case	3.3	A vector appear	N	P
	Point	Point (1,1)			
	Point	Point (2,2)			
DR024	Case	3.5	A ray appear, the ray pass the coordinate (3,3)	N	P
	Point	Point (1,1)			
	Point	Point (2,2)			
DR025	Case	3.6	A line appear, the line pass the coordinate (0,2)	N	P
	Point	Point (1,1)			
	Line	Point (2,3) Point (3,2)			
DR026	Case	3.7	A line appear, the line pass the coordinate (0,0)	N	P
	Point	Point (1,1)			
	Line	Point (2,3) Point (3,2)			
DR028	Case	3.8	Two new lines appear	N	P
	Semicircle	Point (7,0) Point (2,0)			
	Point	Point (3,3)			
DR039	Case	3.8	Two new lines appear	N	P
	Circular Arc	Point (7,0) Point (12,0) Point (0,-1)			
	Point	Point (3,3)			
DR030	Case	3.8	Two new lines appear	N	P
	Circular Arc	Point (4;-4) Point (12,0) Point (2,0)			
	Point	Point (3,3)			
DR031	Case	3.8	Two new lines appear	N	P

	Circle	Point (7,0) Point (2,0)		
	Point	Point (3,3)		
DR032	Case	3.8	Two new lines appear	N P
	Ellipse	Point (6,0) Point (3,0) Point (3,1)		
	Point	Point (3,3)		
DR033	Case	3.8	Two new lines appear	N P
	Parabol a	Line (2,0); (3;1) Point (2,1)		
	Point	Point (3,3)		
DR034	Case	3.8	Two new lines appear	N P
	Hyperb ola	Point (2,0) Point (4,0) Point (1,1)		
	Point	Point (1,3)		
DR035	Case	3.9	A line appear that pass through coordinate (3,1)	N P
	Point	Point (1,1)		
	Point	Point (3,3)		
DR036	Case	3.10	A line appear that pass through coordinate (3,1)	N P
	Segme nt	Point (1,1) Point (3;3)		
	Point	Point (2,-1)		
DR037	Case	3.11	A ray appear that pass through coordinate (2,0)	N P
	Point	Point (1,4)		
	Point	Point (1,1)		
	Point	Point (2,-1)		
DR038	Case	3.12	Two lines appear, one line pass through coordinate (3,1)	N P
	Line	Point (1,1) Point (3;3)		
	Line	Point (0,2) Point (1;-1)		

Draw Curve

Case No.	Field	Input	Output	State	Result
DR039	Case	1.2	Unable to continue step 3	N	P
DR040	Case	3.1	A semicircle appear	N	P
	Point	Point (3,3)			

	Point	Point (1,1)			
DR041	Case	3.2	An arc appear	N	P
	Point	Point (3,3)			
	Point	Point (1,1)			
	Point	Point (1,-1)			
DR042	Case	3.3	An arc appear	N	P
	Point	Point (3,3)			
	Point	Point (1,1)			
	Point	Point (1,-1)			
DR043	Case	3.4	A circle appear	N	P
	Point	Point (3,3)			
	Radius	3			
DR044	Case	3.5	A circle appear	N	P
	Point	Point (3,3)			
	Point	Point (1,1)			
DR045	Case	3.6	A circle appear	N	P
	Point	Point (3,3)			
	Point	Point (1,1)			
	Point	Point (1,-1)			
DR046	Case	3.7	An ellipse appear	N	P
	Point	Point (3,3)			
	Point	Point (1,1)			
	Point	Point (1,-1)			
DR047	Case	3.8	A parabola appear	N	P
	Point	Point (1,3)			
	Line	Point (1,1) Point (3;3)			
DR048	Case	3.9	A hyperbola appear	N	P
	Point	Point (3,3)			
	Point	Point (1,1)			
	Point	Point (1,-1)			

Draw Polygon

Case No.	Field	Input	Output	State	Result
DR049	Case	1.2	Unable to continue step 3	N	P
DR050	Case	3.1	A polygon appears	N	P
	Point	Point (-3,2)			
	Point	Point (3,2)			

	Point	Point (3,1)		
	Point	Point (5,-2)		
	Point	Point (3,-2)		
	Point	Point (2,0)		
	Point	Point (0,0)		
	Point	Point (-1,1)		
	Point	Point (-3,1)		
DR051	Case	3.2	A hexagon appears.	N P
	Point	Point (3,1)		
	Point	Point (1,1)		
	Vertexes	6		
DR052	Case	3.2	Message appear "Please input number higher than or equal 3"	N P
	Point	Point (2,1)		
	Point	Point (1,1)		
	Vertexes	-1		

Draw tangent

Case No.	Figure	Input	Output	State	Result
DR053	Circle	A(0,0)	A tangent is draw and pass through C with equation $x-2=0$	N	P
		B(2,0)			
	Point	C(2,0)			
DR054	Curve	A(0,0)	Two tangent is draw and pass through C	N	P
		B(2,0)			
	Point	F(3,3)			
DR055	Curve	A(0,0)	No tangent is draw	N	P
		B(2,0)			
	Point	C(0,0)			

Translate

Case no.	Figure	Fields	Input	Output	State	Result
TL00 1	Angle	Vector	3,-1	The translated angle is successfully and correctly created with B'(4,3)	N	P
		Angle	A(-2,4)			
			B(1,4)			
			C(1,2)			
TL00 2	Arc	Vector	0,-3	The translated arc is successfully and correctly created with A'(-2,2), B'(-3,0)	N	P
		Arc AB	A(-2,5)			
			B(-3,3)			

TL00 3	Arc	Vector	1,2	The translated arc is successfully and correctly created with A'(-2,3), B'(0,2), C'(0,4)	N	P
		Arc	A(-3,1)			
			B(-1,0)			
			C(-1,2)			
TL00 4	Circle	Vector	(3,1)	The translated circle is successfully and correctly created with equation $(x-2)^2 + (y-4)^2 = 9$	N	P
		Circle ABC	A(-1,3)			
			r = 2			
TL00 5	Ellipse	Vector	(0,-1)	The translated ellipse is successfully and correctly created with equation $0.12x^2 + 0.39y^2 - 0.11xy + 0.16x - 1.89y + 1.32 = 0$	N	P
		Ellipse	A(-2,3)			
			B(3,4)			
			C(3,3)			
TL00 6	Parabola	Vector	(0,-3)	The translated parabola is successfully and correctly created with a': $4y^2 + 8x - 20 = 0$	N	P
		Parabola a	$4y^2 + 8x - 20 = 0$			
TL00 7	Hyperbola	Vector	(1,-2)	The translated hyperbola is successfully and correctly created with equation $-6.65x^2 + 29.35y^2 + 59.83x = 122.42$	N	P
		Hyperbola	A(2,2)			
			B(5,2)			
			C(6,1)			
TL00 8	Line	Vector	-2,2	The translated line is successfully and correctly created with A'(-6,3) B'(-2,4), equation $x - 4y + 18 = 0$	N	P
		Line	A(-4,1)			
			B(0,2)			
TL00 9	Point	Equation	$x - 4y + 8 = 0$			
		Vector	0,3	The translated point is successfully and correctly created with A'(-1,4)	N	P
		Point	A(-1,1)			
TL01 0	Polygon	Vector	0,-3	The translated Polygon is successfully and correctly created with A'(-4,-2), B'(-4,-1), C'(-3,-1), D'(-3,-2)	N	P
		Polygon	A(-4,0)			
		ABCD	B(-4,-1)			
			C(-3,-1)			
			D(-3,0)			
TL01 1	Ray	Vector	3,0	The translated ray is successfully and correctly created with A'(-2,1), B'(0,3) with equation $x - y + 3 = 0$	N	P
		Ray	A(-5,1)			
			B(-3,3)			
		Equation	$x - y + 6 = 0$			

TL01 2	Segment	Vector	-2,-2	The translated segment is successfully and correctly created with A(-4,2), B'(-1,2)	N	P
		Segment	A(-2,4) B(1,4)			
TL01 3	Vector	Vector	1,-1	The translated vector is successfully and correctly created with A'(-1,2), B'(0,1)	N	P
		Vector	A(-2,3) B(-1,2)			

Rotate

Case no.	Figure	Fields	Input	Output	State	Result
TL014	Angle	Angle	45	The rotated angle is successfully and correctly created with B'(-2.83,3)	N	P
		Center	D(0,3)			
		Angle	A(-4,0)			
			B(-2,1)			
			C(-3,2)			
TL015	Arc	Angle	360	The rotated arc is successfully and correctly created and is the same with the original	N	P
		Center	D(0,3)			
		Arc	A(-4,2)			
			B(-3,1)			
			C(-6,3)			
TL016	Circle	Angle	400	The rotated circle is successfully and correctly created with equation: $(x-1)^2 + y^2 = 2,5$	N	P
		Center	D(1,0)			
		Circle	A(-3,2)			
			B(-1,0)			
			C(0,1)			
TL017	Ellipse	Angle	45	The rotated ellipse is successfully and correctly created with equation $0.17x^2 + 0.23y^2 + 0.08xy - 2.66x - 2.13y + 12.1 = 0$	N	P
		Center	D(7,0)			
		Ellipse	A(3,2)			
			B(6,3)			
			C(7,3)			
TL018	Parabola	Angle	-45	The rotated parabola is successfully and correctly created with a' : $0.07x^2 + 19.9y^2 - 2.4xy - 44.6x - 152y + 461.4 = 0$	N	P
		Center	D(5,0)			
		Directrix	A(-2,1)			
			B(2,-1)			
		Parabola a	C(1,1)			
TL019	Hyperbola	Angle	-45	The rotated hyperbola is	N	P

		Center	D(3,0)	successfully and correctly created with equation $4x^2 - 4y^2 + 10.2xy - 9.9x + 12.7y - 7.1 = 0$		
		Hyperbo la	A(-1,3) B(1,3) C(3,4)			
		Line	Angle Center Line AB Equation		The rotated line is successfully and correctly created with the equation - $3.4x + 3.8y - 20.5 = 0$	N P
TL020	Line		60 D(0,3) A(-2,4) B(-1,-1) $-5x - y - 6 = 0$			
TL021	Point	Angle Center Point	-45 D(-1,0) A(-3,2)	The rotated point is successfully and correctly created with A'(-3.83,0)	N P	
TL022	Polygon	Angle Center Polygon ABCD	-450 D(0,0) A(-4,3) B(-5,1) C(-3,1) D(-2,2)			
TL023		Ray	Angle Center Ray AB Equation	The rotated ray is successfully and correctly created with A'(0.73, -3.66), B'(0.91, -0.46) with equation x-y+3=0	N P	
TL024		Segment	Angle Center Segment AB			
TL025		Vector	Angle Center Vector	The rotated vector is successfully and correctly created with A'(-2, -4), B'(-2, -1)	N P	

Reflect

Case no.	Figure	Fields	Input	Output	State	Result
Reflected by Point						
TL026	Angle	Point	D(0,0)	The reflected angle is successfully and correctly created with B'(3, -2)	N	P
		Angle ABC	A(-5,0) B(-3,2) C(-3,0)			

TL027	Arc	Point	D(-1,0)	The reflected arc is successfully and correctly created with A'(0,3), B'(-1,1), C'(0,-1)	N	P
		Arc ABC	A(-2,3) B(-1,1) C(-2,1)			
TL028	Circle	Point	C(0,0)	The reflected circle is successfully and correctly created with equation $(x-2.5)^2 + (y+2.5)^2 = 2.5$	N	P
		Circle ABC	A(-2,4) B(-1,3) C(-3,1)			
TL029	Ellipse	Point	D(-1,2)	The reflected ellipse is successfully and correctly created with equation $0.39x^2 + 0.3y^2 + 0.1xy + 3.3x - 0.7y + 7 = 0$	N	P
		Ellipse	A(2,3) B(3,1) C(4,1)			
TL030	Parabola	Point	D(1,1)	The reflected parabola is successfully and correctly created with equation $x^2 - 2x - 2y + 2 = 0$	N	P
		Directrix	Ox			
		Parabola a	$x^2 - 2x - 2y + 2 = 0$			
TL031	Hyperbola	Point	D(0,3)	The reflected hyperbola is successfully and correctly created with equation $-2.9x^2 - 2.9y^2 - 11.1xy + 58.7x + 43y - 141.7 = 0$	N	P
		Hyperbo la	A(-2,-2) B(-1,1) C(-1,0)			
TL032	Line	Point	D(-2,0)	The reflected line is successfully and correctly created with equation $x - 4y + 4 = 0$	N	P
		Line AB	A(-4,1) B(0,2)			
		Equation	$x - 4y + 8 = 0$			
TL033	Point	Point	D(-2,0)	The reflected point is successfully and correctly created with A'(-4,-2)	N	P
		Point	A(0,2)			
TL034	Polygon	Point	C(0,0)	The reflected Polygon is successfully and correctly created with A'(3,-3), B'(1,-2), C'(0,0), D'(3,0), E'(4,-1)	N	P
		Polygon ABCDE	A(-3,3)			
			B(-1,2)			
			C(0,0)			
			D(-3,0)			
			E(-4,1)			
TL035	Ray	Point	A(0,1)	The reflected ray is successfully and correctly created with A'(0,1), B'(0,4)	N	P
		Ray AB	A(0,1)			
			B(0,-2)			

		Equation	-3x=0	with equation 3x=0		
TL036	Segment	Point	1,-1	The reflected segment is successfully and correctly created with A(2,-3), B'(4,-3)	N	P
		Segment AB	A(0,1)			
			B(-2,1)			
TL037	Vector	Point	B(-1,1)	The reflected vector is successfully and correctly created with A'(1,1), B'(-1,1)	N	P
		Vector AB	A(-3,1)			
			B(-1,1)			
Reflected by Line						
TL038	Angle	Line AB	A(0,2), B(-2,0)	The reflected angle is successfully and correctly created with A'(1,-2), B'(0,-1), C'(-1,0)	N	P
		Angle ABC	A(-2,4)			
			B(1,4)			
			C(1,2)			
TL039	Arc	Line	x-y+1=0	The reflected arc is successfully and correctly created with A'(1,-2), B'(0,-1), C'(-1,-3)	N	P
		Arc ABC	A(-3,2)			
			B(-2,1)			
			C(-4,0)			
TL040	Circle	Line	-y+2=0	The reflected circle is successfully and correctly created with A'(-2,4), B'(-1,4) and equation $(x+2)^2+(y-4)^2=1$	N	P
		Circle ABC	A(-2,0)			
			B(-1,0)			
		Equation	$(x+2)^2+y^2=1$			
TL041	Ellipse	Line	-y+1=0	The reflected ellipse is successfully and correctly created with equation $x^2+0.5y^2+y-0.5=0$	N	P
		Ellipse	A(0,2)			
			B(0,4)			
			C(1,3)			
		Equation	$x^2+0.5y^2-3y+3.5=0$			
TL042	Parabola	Line	-4x-y-3=0	The reflected parabola is successfully and correctly created with a': $15.5x^2+10.5y^2+25.5xy+143.2x+78y+299=0$	N	P
		Parabola a	A(2,1)			
			$-x-5y+4=0$			
TL043	Hyperbola	Line	D(0,2)	The reflected hyperbola is successfully and correctly created with a': -	N	P
			E(7,0)			
		Hyperbo	A(-1,3)			

		Ia		3.1x^2+0.4y^2-1.7xy- 5.3x-5.3y+2.6=0		
			B(-3,1)			
			C(-4,1)			
TL044	Line	Vector	(3,0)	The reflected line is successfully and correctly created with equation -x - y + 3 = 0		N P
		Line AB	A(-1,-1)			
			B(2,2)			
		Equation	x - y = 0			
TL045	Point	Line	2x-y+10=0	The reflected point is successfully and correctly created with A'(-3.4,1.2)	N P	
		Point	A(-5,2)			
TL046	Polygon	Line	-y+1=0	The reflected Polygon is successfully and correctly created with A'(-2,0), B'(-1,0), C'(-1.5,-0.87)	N P	
		Polygon ABCD	A(-2,2)			
			B(-1,2)			
			C(-1.5,2.87)			
TL047	Ray	Vector	(1,0)	The reflected ray is successfully and correctly created with equation -x - 2y-6=0	N P	
		Ray AB	A(-4,1)			
			B(-2,2)			
		Equation	x-2y+6=0			
TL048	Segment	Line	3x-5y+14=0	The reflected segment is successfully and correctly created with A(1.06,2.24), B'(-1.59,3.65)	N P	
		Segment AB	A(0,4)			
			B(0,1)			
TL049	Vector	Vector	(4,0)	The reflected vector (-2,0) is successfully and correctly created with A'(-4,1), B'(-2,0)	N P	
		Vector AB	A(-4,1)			
			B(-2,2)			

Change properties

Case no.	Fields	Input	Output	State	Result
DR011	Tên	A	Object name is A and is invisible	N	P
	Màu sắc	Blue			
	Ẩn/Hiện	Checked			
	Độ dày	1			
	Ẩn/Hiện tên	Uncheck			
	Loại nét	Liền			
DR012	Tên	A	Object name is A, visible without	N	P

	Màu sắc	Blue	label, blue color and invisible with thin dash magrin		
	Ẩn/Hiện	Uncheck			
	Độ dày	1			
	Ẩn/Hiện tên	Uncheck			
	Loại nét	Gạch			
DR013	Tên	A	Object and its name is visible with red color with thick dot-dashed magrin	N	P
	Màu sắc	Red			
	Ẩn/Hiện	Uncheck			
	Độ dày	5			
	Ẩn/Hiện tên	Checked			
	Loại nét	Chấm gạch			

Move (Test on all figure)

Case no.	Input	Result	State	Result
DR014	Choose an object, drag and drop it on screen	Object location is change, other dependent object is redraw	N	P

Remove

Case no.	Input	Result	State	Result
DR015	Choose an object, click "Xóa" button	Object is deleted	N	P
DR016	Choose an object, click Del button	Object is deleted	N	P

Undo (Test on all figure)

Case no.	Input	Result	State	Result
DR017	Click button " \leftarrow "	Return action to previous state in "Quá trình" list	N	P
DR018	Click Ctrl+Z	Return action to previous state in "Quá trình" list	N	P

Redo (Test on all figure)

Case no.	Input	Result	State	Result
DR019	Click button " \rightarrow "	Return action to next state in "Quá trình" list	N	P

DR020	Click Ctrl+Y	Return action to next state in "Quá trình" list	N	P
-------	--------------	---	---	---

Zoom screen

Case no.	Input	Result	State	Result
DR021	Click on button "Phóng to"	Screen zoom in	N	P
DR022	Click on button "Thu nhỏ"	Screen zoom out	N	P
DR023	Uncheck checkbox "Zoom" then scroll up and down	Nothing happen	N	P
DR024	Check checkbox "Zoom" then scroll up and down	Screen zoom in and zoom out	N	P

Move screen

Case no.	Input	Result	State	Result
DR025	Uncheck checkbox "Move" then drag the screen	The screen is immobile	N	P
DR026	Check checkbox "Move" then drag the screen	The center of the screen change	N	P

Software User's Manual

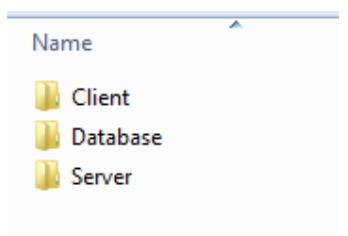
Installation Guide

NOTE:

Before installing, make sure that the server computer already has **SQL Server 2008** and **IIS version 7.0** or newer installed.

This section is intended for web server administrator, who has deep knowledge about IIS (Internet Information Service) and SQL Server.

All necessary files are located in folder Release. You should copy those files and folders to a location on hard disk, in this guide, we use D:\Project\Pythagore. This folder should look like:



D:\Project\Pythagore Folder Content:

1. **Client folder:** all necessary files for web server providing Silverlight client GUI
2. **Database folder:** all necessary files to setup database
3. **Server folder:** all necessary files for WCF web services

Create application pools

You must create 2 new application pools in IIS with the following configuration:

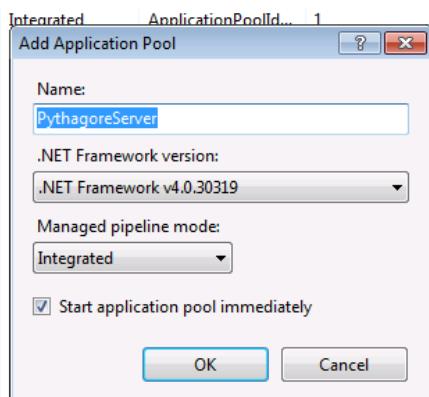


Figure 35 - PythagoreServer Application Pool

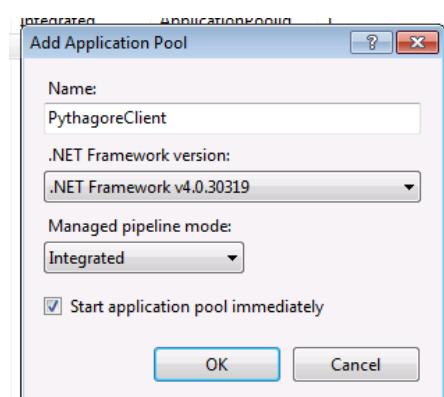


Figure 36 - PythagoreClient Application Pool

Create websites

You must create 2 new websites in IIS with the following configuration

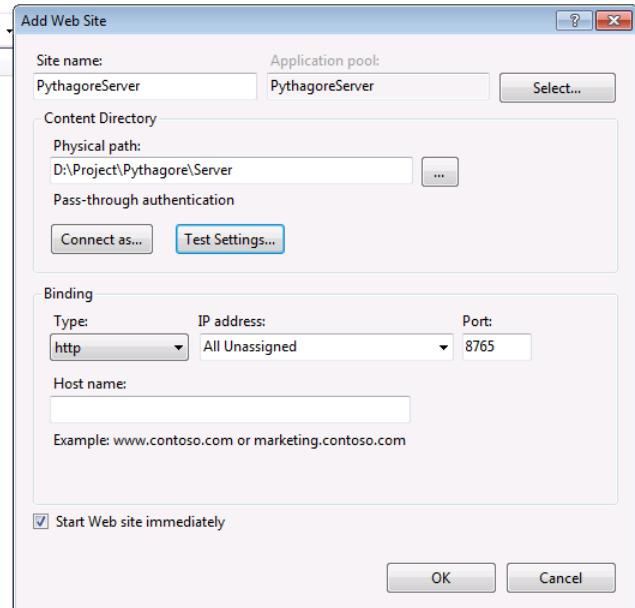


Figure 37- PythagoreServer website configuration

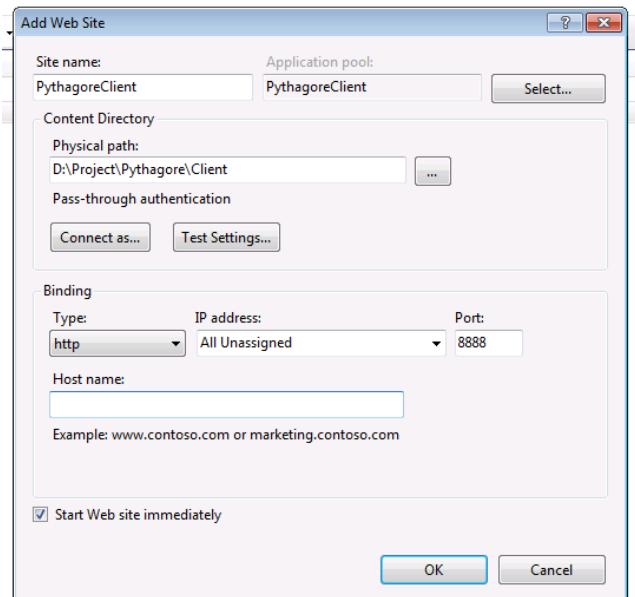


Figure 38 - PythagoreClient website configuration

Binding section varies from actual server configuration.

In this guide, we use port **8888** and **8765** on machine name: **TUNGVV60244**

Note:

If you use a new port number, remember to allow that port in firewall settings

After creating new websites, you should see the following

Name	ID	Status	Binding	Path
Default Web Site	1	Started (ht...)	*:80 (http)	%SystemDrive%\inetpub\wwwroot
PythagoreClient	3	Started (ht...)	*:8888 (http)	D:\Project\Pythagore\Client
PythagoreServer	2	Started (ht...)	*:8765 (http)	D:\Project\Pythagore\Server

Figure 39 - IIS Configuration step complete

Create database in SQL Server Express

Open SQL Server Management Studio, login with your username and password. Then open the file `PythagoreDatabase.sql` in folder **D:\Project\Pythagore\Database** and execute the code.

Take note your connection string, for example:

```
Data Source=TUNGVV60244\SQLEXPRESS;Initial  
Catalog=PythagoreDatabase;Persist Security Info=True;User  
ID=sa;Password=root
```

Configure web.config in Service

Open file **D:\Project\Pythagore\Server\Web.config** using any text editor.

Looking for this line:

```
<add name="PythagoreConnectionString"
```

Change `connectionString` value to the actual connection string (see above)

Save and close that file.

Configure web.config in Client

Open file **D:\Project\Pythagore\Client\Web.config** using any text editor.

Looking for this line:

```
<add key="WCF"
```

Change `value` to the actual hostname and port of PythagoreServer website, for example: `http://tungvv60244:8765/`

Save and close that file.

We have completed installation

User's Guide

Phần 1 – Cấu hình tối thiểu - Minimum System Requirements

Tiếng Việt

Để có thể chạy được trang web một cách mượt mà, bạn cần có một máy tính với cấu hình tối thiểu như sau:

- Vi xử lí 1GHz trở lên
- Máy có RAM 256MB trở lên
- Ổ cứng còn trống 1GB trở lên
- Hỗ trợ màn hình có độ phân giải 1024x768 trở lên
- Hệ điều hành Window XP trở lên
- Máy có thể truy cập internet
- Máy sử dụng trình duyệt Internet Explorer 8.0 trở lên, Firefox 3.0 trở lên, hoặc Chrome 4.0 trở lên.
- Trình duyệt của bạn phải được cài thêm plugin Silverlight

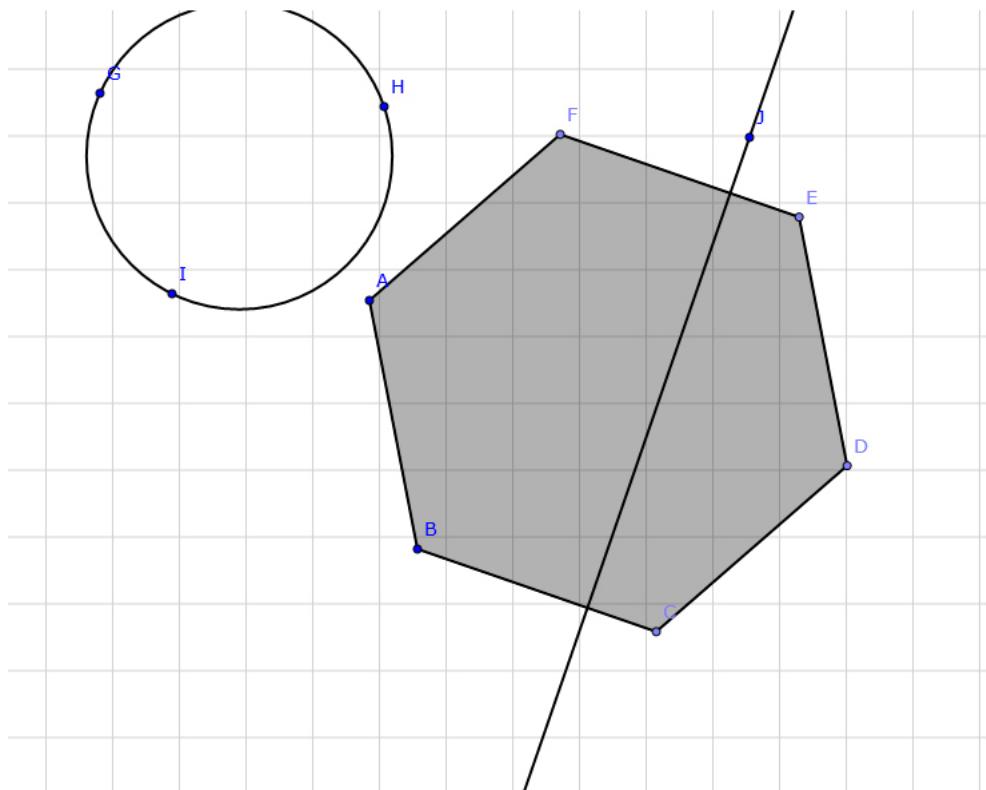
English

To run our website smoothly, you need a computer with:

- 1GHz CPU or above
- 256MB RAM or above
- 1GB space on HDD or above
- Screen resolution need to be 1024x768 or higher
- Window XP or an upper version
- Have an internet connection
- Have at least one of these browser: Internet Explorer 8.0 or later, Firefox 3.0 or later, Chrome 4.0 or later
- Your browser needs the Silverlight plugin.

Phần 2 – Giới thiệu về trang web - Introduction to the website

- Pythagore là một phần trong dự án e-school tại Việt Nam của chúng tôi, là một công cụ hỗ trợ người giảng dạy môn hình học cấp 2 có thể hiện thực và truyền tải đúng những gì mà chương trình yêu cầu chỉ bằng máy tính và chuột, không cần phải đo và vẽ bằng tay nữa.
- Trang web hỗ trợ gần như đầy đủ các công cụ mà một giáo viên cần có, bao gồm tạo 1 lớp học với số lượng học sinh cần thiết, tạo nhiều bài cùng một lúc, có quyền nói chuyện, mời học sinh vẽ hoặc mời học sinh ra khỏi phòng. Trang web còn hỗ trợ cho học sinh chúc năng giơ tay khi muốn phát biểu ý kiến.
- Pythagore is a part of our 'e-school n Vietnam' project, a tool to support Geometry teacher in junior high school to draw and show the student exactly what the lesson required just by computer and mouse, no more hand-drawing on blackboard.
- The site supports virtually all the tools that a teacher needs: creating a class with a number of students needed, creating multiple sheets at once, having the right to speak, giving student the permission to draw or kicking them out of the room. The site also supports student to raise hands when they want express their opinions.



- Với chức năng vẽ của trang web, người dùng có thể vẽ được hầu hết những hình mà chương trình hình học cấp 2 yêu cầu. Trong 1 khoảng thời gian, chỉ có một người trong 1 phòng được cấp quyền vẽ, quyền này được quyết định bởi người tạo phòng (giáo viên).
- With the drawing function of the site, users can draw almost everything that the junior high school geometry may require. In a period of time, only one person has the permission to draw on the pad, this permission is decided by the creator of room, usually the teacher.



- Trang web còn hỗ trợ việc chia sẻ bài đang làm với một trang web khác, hoặc việc tải trang về, tải trang lên, chụp lại hình trang hiện tại nhằm giúp cho việc giáo viên soạn giáo án dễ dàng hơn.
- Our website also support user to share the sheet with other page, download the sheet, upload another sheet or capture the current sheet, this will help the teacher in planing their lesson much easier.

Phần 3 – Các chức năng và cách sử dụng – Features and how to use

Tài Khoản – Account

- Mục này sẽ hướng dẫn bạn cách quản lý tài khoản của mình, bao gồm Đăng nhập, Đăng ký mới hoặc thay đổi thông tin cá nhân.

In this section, we will guide you how to manage your account, including Sign up, Sign in or change your information.

- **Đăng Nhập – Login**

The image shows a login form titled 'Đăng nhập' (Login). It has two input fields: 'Tên đăng nhập' (Username) and 'Mật khẩu' (Password), both with grayed-out placeholder text. Below the fields is a link 'Chưa đăng ký?' (Not registered?). At the bottom are two buttons: a teal-colored 'Đăng nhập' (Login) button and a white 'Hủy' (Cancel) button.

- Đây là trang mặc định sẽ hiện ra khi bạn truy cập vào trang web, nếu bạn đã có tài khoản trang web, các bạn chỉ cần nhập tên truy cập và mật khẩu vào và chọn Đăng nhập. Nếu bạn chưa có tài khoản, hãy chọn "Chưa đăng ký?" để chuyển đến trang đăng ký tài khoản mới.

This is the default page, which will appear right after you open the site, if you already have an account, please enter your username and password then click 'Đăng nhập'. If you don't have any account, please click "Chưa đăng ký?" link to register you a free account.

- **Đăng ký mới – Register**

The image shows a registration form titled 'Đăng ký' (Register). It has four input fields: 'Tên đăng nhập' (Username), 'Mật khẩu' (Password), 'Nhập lại mật khẩu' (Re-enter password), and 'Địa chỉ mail' (Email address), all with grayed-out placeholder text. Below the fields is a link 'Đã đăng ký? Chọn vào đây để đăng nhập!' (Already registered? Click here to log in!). At the bottom are two buttons: a teal-colored 'Đăng ký' (Register) button and a white 'Hủy' (Cancel) button.

- Tại đây bạn có thể đăng ký tài khoản mới để truy cập vào trang web. Bạn cần nhập đầy đủ vào 4 ô được hiển thị bao gồm: Tên đăng nhập, mật khẩu, xác nhận mật khẩu, thư điện tử. Sau đó chọn "Đăng Ký", bạn sẽ được chuyển lại về trang Đăng nhập.
- In this page, you can register yourself a new account. You need to enter your information including: Username, password, retype password, and your email. After fill all the box, now click "Đăng Ký", you will be redirect to Login page.

- **Thoát – Exit**



- Bạn hãy thoát khỏi tài khoản của mình khi không còn sử dụng trang web nữa, tất cả thông tin trong quá trình bạn tham gia trang web sẽ được lưu lại cho lần sau. Từ giao diện trang chủ, bạn chọn Thoát trên góc phải.
- If you don't use this site anymore, please log out your account, all information has been stored for your later visit. From home page, click 'Thoát' at the up right corner.

- **Thay đổi thông tin cá nhân – Change information**

- Sau khi đăng nhập vào tài khoản, bạn có thể thay đổi thông tin cá nhân của mình bằng cách chọn 'Thông tin cá nhân' trên góc phải. Tại đây, bạn có thể thay đổi tên hiển thị, thư điện tử hoặc mật khẩu. Sau khi thay đổi xong, bạn hay chọn "Lưu".
- After logged in, you can change your information easily, in Home page, click 'Thông tin cá nhân'. In this page, you can change you Display name, email and your password. After changed your information, you need to click "Lưu" to save it.

Danh sách liên lạc – Contact

Đây là mục hướng dẫn bạn cách sử dụng Danh sách liên lạc.

This session will guide you how to use your contact list:

- **Giới thiệu về danh sách liên lạc - Overview of Contactlist**

Danh sách liên lạc



- Danh sách liên lạc được đặt phía bên trái của trang chủ ngay sau khi bạn đăng nhập thành công.
- Danh sách liên lạc được chia theo danh sách nhóm, bạn có thể tạo nhiều nhóm, trong mỗi nhóm có nhiều tên của liên lạc đã được thêm vào. Bạn có thể thu gọn hoặc mở nhóm ra để xem.
- Mỗi liên lạc được hiển thị gồm có. Một kí hiệu cho biết người đó có đang online hay không, Tên, và một nút dung để xóa người đó khỏi danh sách (chỉ hiện khi bạn rê chuột vào tên)
- Contact list is placed in the left panel of home page.
- This list is divided by group, you can have many groups, in each group there are many other contacts. You can collapse or open group to view its content.
- Each contact contains one icon to show if user is online, name, and an delete button, which only appear when you hover the name.

- **Thêm liên lạc vào danh sách – Add contact**

Danh sách liên lạc

[Thêm liên lạc](#) [Thêm nhóm](#)

Thêm liên lạc

Tên liên lạc

Tung vv

Nhóm liên lạc

nhóm 2

[Thêm](#)

[Hủy](#)

- Để thêm liên lạc vào danh sách, trước hết bạn cần có ít nhất một nhóm trong danh sách, nếu chưa có nhóm, bạn sẽ được tự động chuyển đến trang thêm nhóm.
- Từ trang chủ, chọn “Thêm liên lạc” từ Danh sách liên lạc. Sau đó nhập vào Tên Liên lạc và chọn nhóm để thêm vào. Chọn ‘Thêm’ để hoàn tất.
- To add contact, you need to have at least one group, if you don't have any, you will be redirect to add group page.
- From home page, click “Thêm liên lạc”. You need to enter contact's name and chose on group to place that contact in. Click ‘Thêm’

- **Thêm nhóm – Add group**

Danh sách liên lạc

[Thêm liên lạc](#) [Thêm nhóm](#)

Thêm nhóm

Tên nhóm

[Thêm](#)

[Hủy](#)

- Chọn “Thêm nhóm”
- Nhập vào tên nhóm và chọn “Thêm”.
- From home page, click “Thêm nhóm”.
- Enter group name, then click “Thêm”.

- **Xóa liên lạc khỏi danh sách – Delete contact**



- Để xóa một bạn trong danh sách, bạn rê chuột vào tên người đó, nhấn vào nút vừa 'X' hiện ra sau tên, sau đó bấm 'OK' để xóa.
- To delete a contact, hover on the contact name, an 'X' button will appear, click that button, then click "Ok" for confirmation.

- **Nhận lời mời thêm liên lạc – Accept contact**

- Khi bạn được người khác mời thêm liên lạc, bạn sẽ nhận được bảng thông báo là ai đó gửi lời mời, bạn chọn OK để hoàn tất quá trình thêm liên lạc.
- Someone want to add you to their contact list, you will be informed about this when you log in, just click OK to finish.

Quản lý trang chủ - Home screen management

- Quản lý phòng – Room management:
- Xem phòng – View room



- Sau khi bạn đã đăng nhập vào, màn hình trang chủ sẽ xuất hiện. Tại đây, bạn có thể nhìn thấy toàn bộ những phòng bạn đã tạo trong khu vực "Phòng của bạn" và những phòng bạn được mời trong khu vực
- After you have logged in, a new screen appears. Here, you can find all your created rooms, invited rooms or pending rooms. All of your created rooms will be stored in "Phòng của bạn" area.

"Phòng được mời".

- Những phòng bạn từng vào rồi sẽ có khung nền xanh dương, và nếu chưa thì có khung nền màu cam. Trong mỗi cái khung của mỗi phòng, số trang, tên phòng và tên người tạo phòng sẽ được thể hiện ra trên đó.
- Để xem một căn phòng cụ thể, bạn hãy nhấp chuột trái vào ngay cái phòng đó.

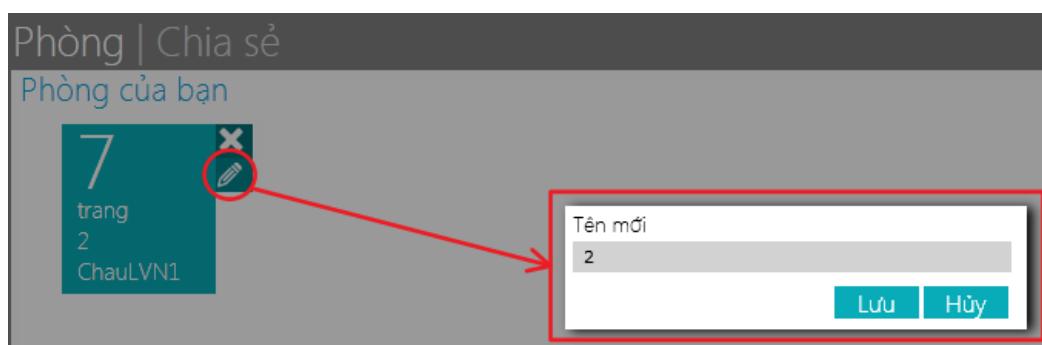
- **Xóa phòng – Delete room**



- Để xóa căn phòng, bạn hãy bấm vào dấu "X" nằm ở góc trên bên phải của phòng cần xóa. Phòng bị xóa sẽ biến mất khỏi khu vực phòng của bạn.
- Chú ý: Nếu bạn bấm vào dấu "X" của phòng bạn được mời, phòng này sẽ chỉ biến mất mà không bị xóa. Lúc này, bạn chỉ bị xóa khỏi danh sách người tham dự của phòng đó thôi.

- All of your pending rooms will be stored in "Phòng được mời" area. Any room that you have entered will display in blue background button, otherwise, they will display in orange background button. The button contains information include: Number of sheets, the room name, and the room owner.

- **Đổi tên phòng – Rename room**



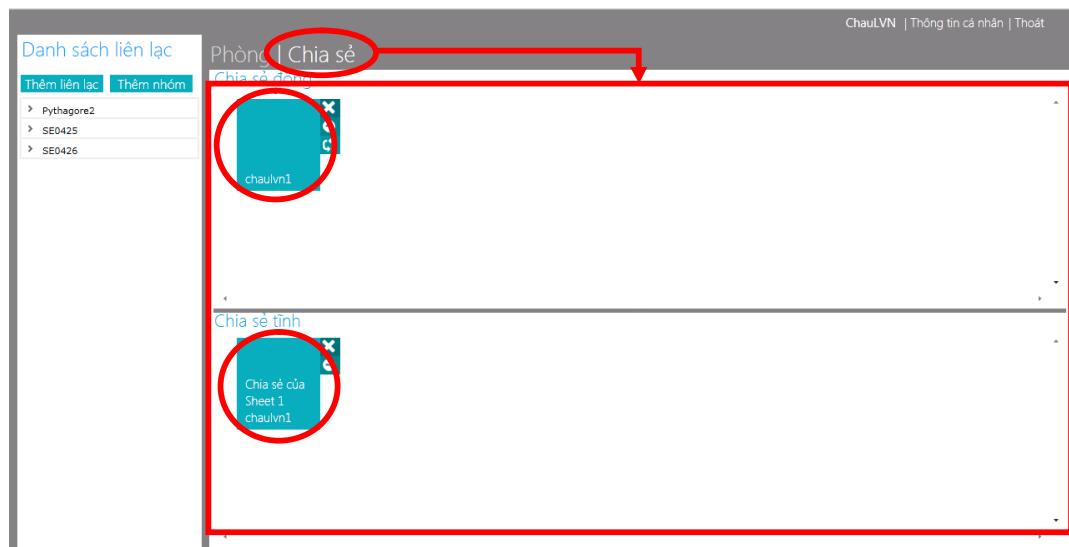
- Để đổi tên căn phòng, bạn hãy bấm vào hình cây bút chì nằm ở góc trên bên phải, dưới dấu "X" của phòng cần đổi tên. Một hộp thoại sẽ hiện ra. Lúc này bạn nhập tên mới của căn phòng vào trong khu vực "Tên mới" và bấm "lưu" để hoàn tất. Hoặc chọn "hủy" nếu bạn không muốn đổi tên.
- Chú ý: Bạn không thể đổi tên của phòng được mời.

- **Thêm phòng – Add room**



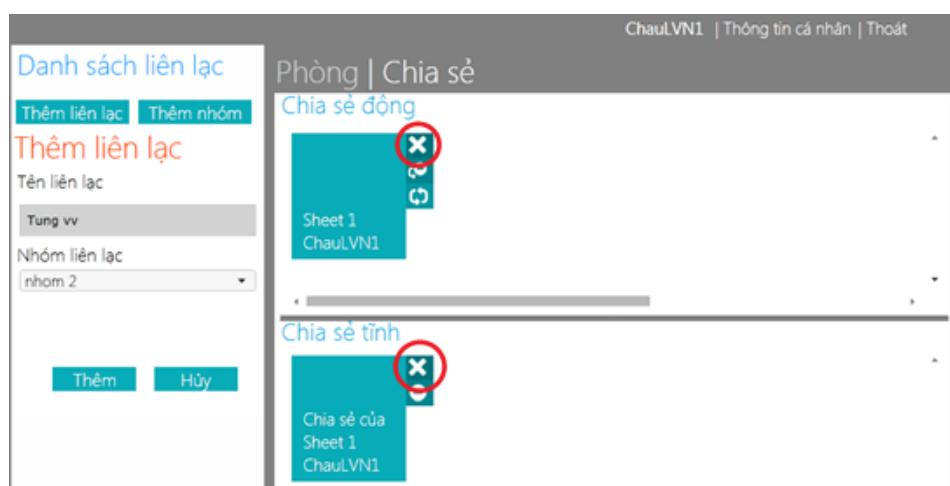
- Để tạo thêm phòng mới, bạn hãy bấm vào nút "Thêm Phòng" nằm ở góc trên bên phải của khu vực "Phòng của bạn". Sau khi bấm xong, bạn hãy điền tên phòng và chọn OK để hoàn tất việc tạo phòng.
- To add new room, left click "Thêm phòng" button when you are in home screen (the screen after you logged in). After clicked, type the name of the room, and press OK button after that to finish the job.

- **Quản lý chia sẻ - Share management**
- **Xem chia sẻ - View share**



- Từ màn hình chính, bạn chọn thẻ "Chia sẻ". Tại đây, bạn có thể nhìn thấy toàn bộ những chia sẻ động trong khu vực "Chia sẻ động" và những chia sẻ tĩnh trong khu vực "Chia sẻ tĩnh".
- Để xem một chia sẻ cụ thể, bạn hãy nhấp chuột trái vào ngay chia sẻ đó.

- **Xóa chia sẻ - Delete share**



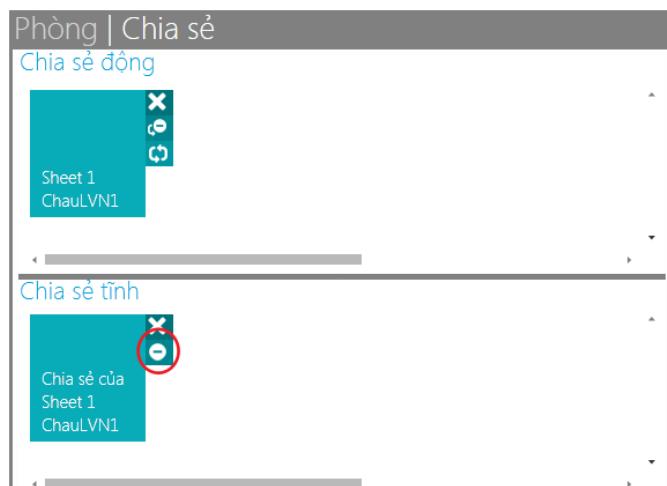
- Tại thẻ "Chia sẻ", bạn hãy bấm vào dấu "X" nằm ở góc trên bên phải của chia sẻ cần xóa. Chia sẻ bị xóa sẽ biến mất khỏi khu vực chia sẻ của bạn.
- From tab "Chia sẻ", you can see all live shares in "Chia sẻ động" area and static shares in "chia sẻ tĩnh" area. To view share link, left click on largest button.

- **Hồi phục chia sẻ bị ngừng – Revert share**



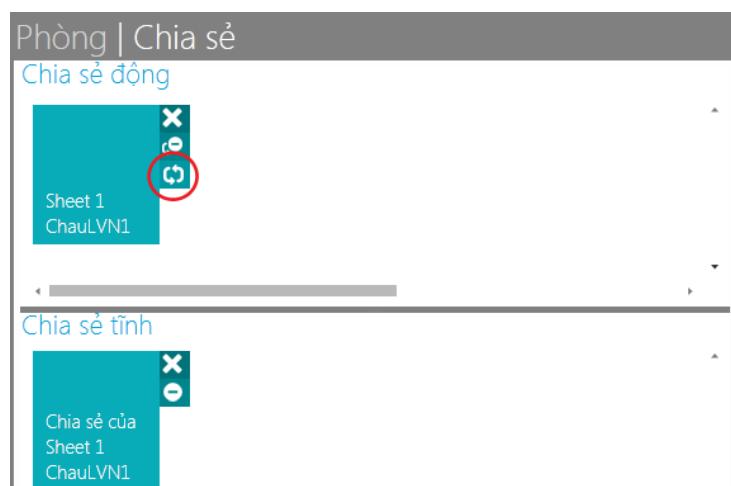
- Tại thẻ "Chia sẻ", những chia sẻ nào đã dừng, thay vì nút ngừng chia sẻ (hình giống bảng cấm đi ngược chiều) thì sẽ xuất hiện nút hồi phục chia sẻ.
- To reverse a share that has been stopped, left click on the button below the X button of the share you want to reverse.
- Notice: this button only appears if you have stopped that share.

- **Ngừng chia sẻ - Stop share**



- Chọn thẻ "Chia sẻ".
- Để ngưng chia sẻ, bạn bấm vào nút ngừng chia sẻ (hình giống bảng cấm đi ngược chiều).
- Make sure you are in the share tab panel.
- To stop a share from displaying in other website, left click on the button below the X button of the share you want to stop.

- **Đổi kiểu chia sẻ - Convert share**



- Chọn thẻ "Chia sẻ".
- Make sure you are in the share tab

- Để chuyển đổi chia sẻ động thành chia sẻ tĩnh, bạn bấm vào nút dưới cùng trong ba nút nhỏ bên phải.
- Chú ý: Chỉ có những chia sẻ nằm trong khu vực chia sẻ động mới có nút này.
- panel.
- To convert a share from live to static, click on the third button in the right side of the share that you want to convert.
- Notice: This only work with live share.

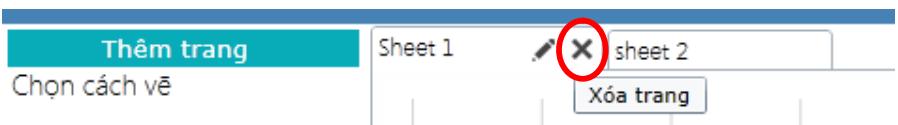
Quản lý trang vẽ - Sheet management

- Thêm trang – Add sheet



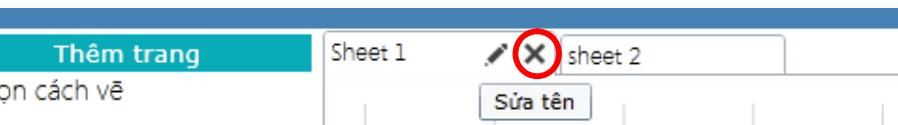
- Đầu tiên, bạn cần phải là chủ một phòng và đang ở trong phòng đó trước khi thực hiện các bước kế tiếp.
- Để tạo thêm trang mới, bạn hãy bấm vào nút "Thêm trang" nằm ở góc trên bên trái của cửa sổ. Sau khi bấm xong, bạn hãy điền tên trang và chọn dấu "+" để hoàn tất việc tạo trang.
- Make sure you are owner of the room screen you're in.
- To add new sheet, left click on the "Thêm trang" button on the top left corner of the screen, a textbox will appear, allow you to type sheet name. After that, press "+" button to add sheet.

- Xóa trang – Delete sheet



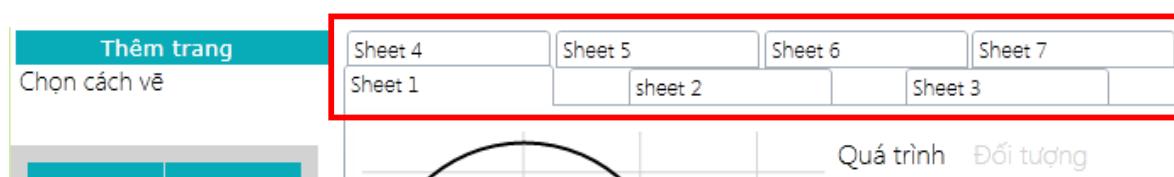
- Đưa con trỏ vào tên trang cần xóa, hai nút nhấn sẽ xuất hiện. Chọn nút nhấn hình dấu X. Khi cần xác nhận, chọn "Có".
- Lưu ý: Khi xóa trang cuối cùng của phòng thì phòng sẽ tự động tạo trang mới.
- To remove a sheet, simply left click on the "X" button on the right of the sheet name. To make the "X" button appear, move the mouse over sheet name.

- Đổi tên trang – Rename sheet



- Đưa con trỏ vào tên trang cần xóa, hai nút mới sẽ xuất hiện. Chọn nút nhấn hình cây bút chì. Sau khi nhập tên mới vào, bạn chọn đồng ý.
- To rename a sheet, move the mouse over sheet name, click on the appeared pencil button on the right of the sheet name. A textbox will appear, allow you to type the new name.

- **Đổi trang hiện hành – Change current sheet**



- Để chuyển sang khác, nhấp chuột trái vào thẻ mang tên trang đó.
- Chú ý: Trong trường hợp số trang nhiều thì các thẻ sẽ nhảy lên nhau.
- To change current sheet, simply left click on the sheet name on top page.
- Notice: Sheet may jump on each other if there are many sheets

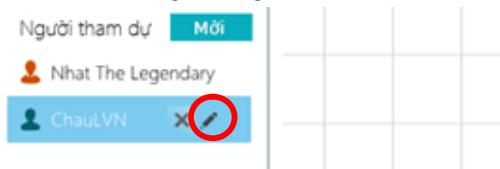
- **Người tham dự - Participant**

- **Mời thêm người tham dự - Invite participant**



- Để mời người dùng vào phòng, bạn nhấp vào nút "Mời" nằm ở bên trái màn hình. Sau đó, bạn nhập tên người dùng bạn muốn mời vào. Bấm dấu "+" để hoàn tất việc mời.
- Chú ý: Hệ thống có hỗ trợ việc tự động kiểm người dùng mà bạn đã thêm vào trong sổ liên lạc.
- To invite another member, left click on "Mời" button. A textbox appear, allow you to type the name of the member you want to invite. Finish by left click on "+" button.
- Notice: Add contact make inviting participant much easier.

- **Đuối người tham dự - Remove participant**



- Để đuối người tham dự trong phòng, bạn đưa chuột vào tên người đó trong danh sách tham dự. Khi có hai ký hiệu hiện ra bên phải tên người đó, bạn hãy chọn chữ
- To remove a participant, left click on the "X" button on the right of participant name. Move the mouse over participant name to reveal "X" button.

"X".

- Chú ý: Người bị đuổi sẽ tự động chuyển về trang lỗi và không thể vào lại phòng.
- Notice: Once removed, the participant can't join back until invited.

- **Trao quyền vẽ - Grant permission**



- Để trao quyền vẽ cho thành viên khác, bạn kéo chuột vào tên thành viên đó ở trong danh sách người tham dự, rồi nhấp vào cây bút chì bên phải tên người đó để trao quyền vẽ.
- Chú ý: sau khi trao, bạn không được vẽ, di chuyển hay thực hiện lệnh quay lại cho đến khi bạn lấy lại quyền vẽ.
- To grant a participant permission to draw, left click on the pencil button on the right of the participant's name. Move the mouse over the participant's name to reveal the pencil button.
- Until you take your permission back, you cannot create new figures or move existing ones.

- **Lấy quyền vẽ - Take permission**



- Để lấy lại quyền vẽ, bạn kéo chuột vào chính tên bạn ở trong danh sách người tham dự, rồi nhấp vào cây bút chì bên phải tên bạn để lấy lại quyền vẽ.
- To get the permission to draw back, left click on the pencil button on the right of your name. Move the mouse over your name to reveal the pencil button.

- **Thanh công cụ - Toolbar**



- Thanh công cụ nằm ở phía dưới của màn hình vẽ, nó cung cấp những tính năng độc đáo trong suốt quá trình vẽ.
- Sau đây là phần giải thích chi tiết:
- The toolbar is right at the bottom of the page. Here several tools that aid drawing lie here.
- This is the detail information of each icon:

Table 7 - Toolbar Icon Description

Features	Icon	Tiếng Việt	English
Chụp ảnh – Screen capture		<ul style="list-style-type: none"> - Chức năng chụp ảnh giúp bạn có thể lưu lại những gì hiện trên khung vẽ về máy của mình. - Chú ý: Chương trình chỉ hỗ trợ định dạng jpeg. 	<ul style="list-style-type: none"> - Allow capturing of the drawing table screen. - Notice: Only support JPEG format.
Tải trang vẽ - Download		<ul style="list-style-type: none"> - Chức năng tải về giúp bạn có thể tải nội dung trang vẽ và sử dụng nó cho trang vẽ khác. 	<ul style="list-style-type: none"> - Allow downloading of the page content for further use.
Tải trang lên - Upload		<ul style="list-style-type: none"> - Chức năng tải lên giúp bạn có thể hồi phục lại nội dung của bài học mà bạn đã tải về. 	<ul style="list-style-type: none"> - Allow uploading of a downloaded page.
Phóng to/Thu nhỏ - Zoom		<ul style="list-style-type: none"> - Chức năng phóng to, thu nhỏ giúp bạn có thể phóng to hoặc thu nhỏ màn hình vẽ. 	<ul style="list-style-type: none"> - Allow zoom in and zoom out of drawing table.
Hiển thị trực, lưới – Show axis, grid		<ul style="list-style-type: none"> - Chức năng hiển thị trực và lưới giúp bạn có thể lựa chọn hiển thị những trợ giúp vẽ như trực và lưới vẽ trên màn hình. 	<ul style="list-style-type: none"> - Allow displaying or hiding axes and gridlines which can aid the drawing.
Giơ tay – Raise hand		<ul style="list-style-type: none"> - Chức năng giơ tay giúp bạn có thể yêu cầu người chủ phòng trao quyền vẽ. 	<ul style="list-style-type: none"> - Allow raising hand to gain the attention of room owner.
Chia sẻ - Share		<ul style="list-style-type: none"> - Chức năng chia sẻ giúp bạn tạo chia sẻ động hay tĩnh của bài học. 	<ul style="list-style-type: none"> - Allow creating live and static share.

- Chức năng chat – Chat Feature
- **Hộp thoại chat - Phóng to – Maximize textbox**



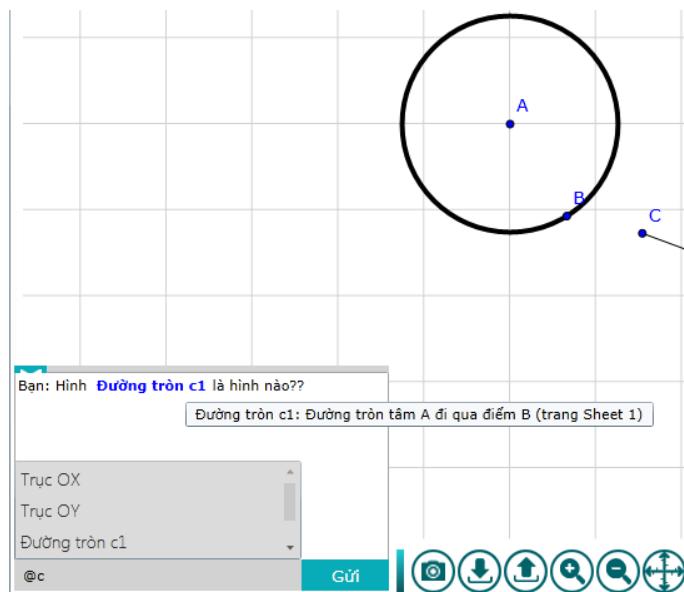
- Pythagore cung cấp cho người dùng hộp thoại chat với chức năng có thể thu gọn hoặc phóng to thuận lợi cho việc tiết kiệm tối đa không gian vẽ.
- Pythagore provides user a Chatbox that can be minimize and maximize, which is a convenience for saving drawing space.

- **Hướng dẫn cách chat – How to chat**



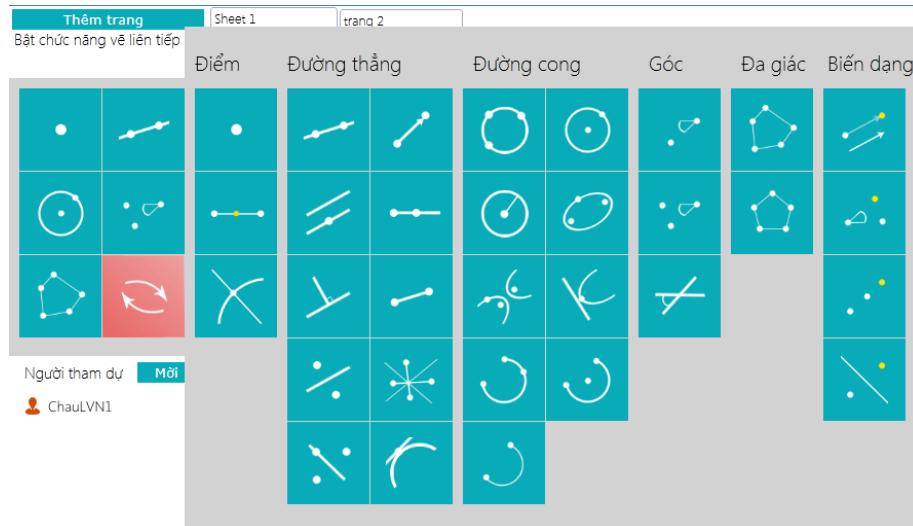
- Để truyền tin nhắn, người dùng gõ tin nhắn vào ô chat, rồi bấm "Gửi". Tin nhắn sẽ tự động được truyền tới những thành viên khác trong phòng.
- Ngoài hỗ trợ người dùng gửi tin nhắn bằng tiếng Việt, Pythagore còn hỗ trợ chức năng tìm nhanh, giúp người dùng dễ dàng chọn hình mình muốn để cập đến trong khi chat bằng cách gõ @Tên hình (vd: @a).
- To send message, after typing into the chat textbox, User clicks the button "Gửi". The message will be automatically sent to all the other participants in that room.
- Besides supporting user to type in Vietnamese, Pythagore also provides "Refer to Object" feature. This means if user wants to refer to an object on the drawing panel, they can type "@Object's Name", Pythagore will suggest all available object with the name mentioned

- **Hỗ trợ tooltip khi rê chuột lên tên đối tượng - Hover on link in Chatbox**



- Người dùng có thể dễ dàng tìm ra hình được đề cập đến trong khi chat bằng cách rê chuột lên tên hình, hình sẽ tự sáng lên hoặc đổi màu, đồng thời cũng sẽ xuất hiện tooltip cho người dùng biết thêm thông tin về hình đang chọn (trang, điểm phụ thuộc, tên hình...).
- User can easily find the mentioned object by hovering the mouse pointer on the object name in the chat box. The referred object will be highlighted or will change color. A tooltip will also appear with object's information (i.e. which sheet the object is on, object's equation or magnitude...).

- **Chức năng vẽ**
- **Hộp công cụ**



- Pythagore cung cấp cho người dùng một hộp công cụ với đầy đủ các loại hình học phục vụ cho việc vẽ và giải các bài tập thuộc hình học phẳng. Hộp công cụ cũng được thiết kế có thể bung ra hoặc thu gọn vào nhằm tiết kiệm không gian vẽ.
- Khi vẽ và cần chọn những đối tượng có sẵn, Pythagore hỗ trợ người dùng chức năng làm sáng lên nhưng đối tượng có thể được chọn, giúp cho việc chọn lựa được dễ dàng (vd: khi cần chọn đường thẳng cần vẽ đường thẳng song song thì khi đó các đối tượng thẳng sẽ sáng lên).
- Ngoài ra, để tiện lợi cho việc vẽ liên tiếp mà không cần chọn lại hình mình muốn vẽ trong hộp công cụ, Pythagore cho phép ta chỉ chọn hình cần vẽ một lần, sau đó sẽ vẽ hình đó liên tiếp đến khi ta bấm Escape hoặc chọn lại hình đó để dừng.
- Sau đây là bảng giải thích chi tiết từng biểu tượng và cách sử dụng của chúng:
- Pythagore provides a full featured drawing toolbox that can be minimized or maximized to save the drawing space. The toolbox has all necessary figures to help user to draw and solve any problems in geometry.
- When user needs to choose existed object, the highlight feature will be very handy. It highlights all the selectable objects on the drawing panel. For example, when user choose to draw a parallel line of segment a, all the linear object will be highlighted.
- In addition, for convenient use of continuous drawing, Pythagore allows user to choose the figure only one time, then continue to draw that figure until the Esc button is pressed, or user choose that figure again.
- The meaning of the icons and their usages will be explained in the following table:

Table 8 - Drawing Toolbox Icon Description

STT / No.	Tên hình/Figure name	Biểu tượng /Icon	Cách vẽ/How to draw
1	Điểm tự do		Chọn một điểm bất kì có sẵn trên bảng vẽ, hoặc chọn điểm mới
	Free point		Choose anywhere or choose an existed point on the drawing pad
2	Trung điểm		Chọn một đoạn thẳng hoặc chọn hai điểm cần vẽ trung điểm
	Midpoint		Choose an existed segment or two points that need to draw a midpoint
3	Giao điểm		Chọn hai hình cần tìm giao điểm: <ul style="list-style-type: none"> • Giao điểm giữa đường thẳng và đường cong • Giao điểm giữa 2 đường thẳng • Giao điểm giữa 2 đường cong

	Intersection		Choose two figures that need to find intersection: <ul style="list-style-type: none">• Two lines• Two curves• A line and a curve
4	Véc tơ đi qua hai điểm		Chọn hai điểm véc tơ đi qua. Điểm đầu tiên là gốc của véc tơ
	Vector formed by two points		Choose two points. The first point will be the tail of the vector
5	Tia đi qua hai điểm		Chọn hai điểm bất kì hoặc hai điểm có sẵn. Điểm đầu tiên sẽ là gốc của tia.
	Ray formed by two points		Choose two points. The first point will be the root of the ray
6	Đường thẳng qua hai điểm		Chọn hai điểm bất kì hoặc hai điểm có sẵn
	Line formed by two points		Choose two points
7	Đoạn thẳng qua hai điểm		Chọn hai điểm bất kì hoặc hai điểm có sẵn
	Segment formed by two points		Choose two points
8	Đường thẳng song song		Chọn một đường thẳng cần vẽ đường song song và một điểm mà đường thẳng song song đó sẽ đi qua.
	Parallel line		Choose a line then a point that the parallel line will pass through
9	Đường thẳng vuông góc		Chọn một đường thẳng và một điểm mà qua đó sẽ vẽ đường vuông góc.
	Perpendicular line		Choose a line and then a point that the perpendicular line will pass through
10	Đường trung trực của hai điểm		Chọn hai điểm muốn vẽ trung trực
	Perpendicular Bisector of two points		Choose two points to draw a bisector
11	Đường trung trực của đoạn thẳng		Chọn đoạn thẳng muốn vẽ trung trực
	Perpendicular		Choose a segment to draw a bisector

	Bisector of a segment		
12	Phân giác của ba điểm		Chọn ba điểm để tạo góc cần vẽ phân giác, điểm giữa sẽ là điểm mà đường phân giác đi qua.
	Angle Bisector of three points		Choose three points. The second point will be the point that the bisector pass through
13	Cặp phân giác của hai đường thẳng		Chọn hai đường thẳng. Pythagore sẽ vẽ cặp phân giác cho hai đường thẳng đó.
	Angle Bisectors of two lines		Choose two lines. Pythagore will draw a pair of bisectors of two chosen lines
14	Tiếp tuyến của đường cong		Chọn một đường cong và một điểm tiếp hạn một đường cong
	Tangent of a curve		Choose a curve and a point through with the tangent will pass
15	Đường tròn ngoại tiếp ba điểm		Chọn ba điểm đường tròn sẽ đi qua
	Circle formed from three points		Choose three point
16	Đường tròn biết tâm và bán kính		Chọn một điểm làm tâm và nhập bán kính
	Circle formed from a center and radius		Choose a center enter a radius
17	Đường tròn biết tâm và đi qua một điểm		Chọn tâm và một điểm đường tròn đi qua
	Circle formed from a center and a point		Choose two points: a center and a point on circle
18	Hình ê líp biết hai tâm và đi qua một điểm		Chọn hai tâm và một điểm thuộc ê líp
	Ellipse formed from two focuses and a point		Choose three points: two focuses and a point on ellipse

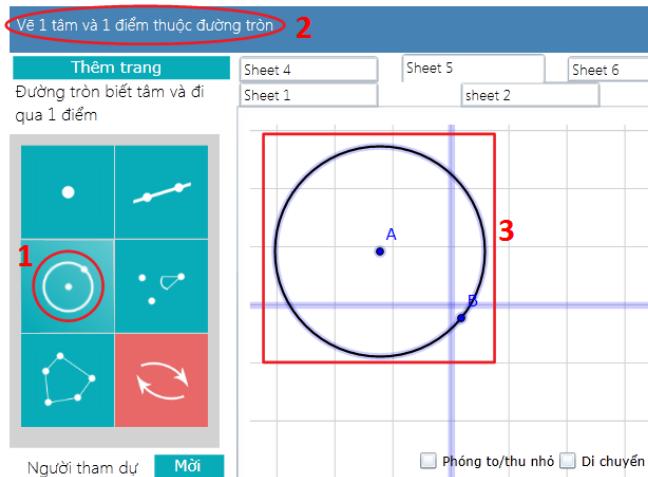
19	Hình hypêbôn biết hai tâm và đi qua một điểm		Chọn hai tâm và một điểm thuộc hypêbôn
	Hyperbola formed from two focuses and a point		Choose three points: two focuses and a point on hyperbola
20	Hình parabol biết tâm và đường chuẩn		Chọn tâm và đường chuẩn
	Parabola formed from a point and a directrix		Choose a point and a linear object as directrix
21	Cung ngoại tiếp ba điểm		Chọn ba điểm cung đi qua
	Circumcircular Arc		Choose three points on arc
22	Cung tạo bởi ba điểm		Chọn tâm và hai điểm trên cung
	Circular Arc		Choose three points: a center and two points on arc
23	Hình bán nguyệt		Chọn hai điểm đầu của cung bán nguyệt
	Semicircle		Choose two points
24	Góc với số đo cho trước		<p>Vẽ góc với điều chỉnh nhập liệu</p> <p>Nhập số đo góc 1 <input type="text" value="90"/> 2 <input checked="" type="radio"/> Theo chiều kim đồng hồ <input checked="" type="radio"/> Ngược chiều kim đồng hồ 3 <input checked="" type="radio"/> Độ <input type="radio"/> Radian <input type="checkbox"/> x n</p> <p>4 OK Hủy</p>
			<p>Chọn hai điểm và nhập góc muốn vẽ</p> <ol style="list-style-type: none"> 1. Góc muốn vẽ 2. Chọn vẽ theo chiều kim đồng hồ hoặc ngược chiều kim đồng hồ 3. Chọn đơn vị tính số đo góc là độ hay radian 4. Bấm "OK"

	Angle with a given magnitude		Choose two points and enter the magnitude of the angle 1. Magnitude of the angle 2. Clockwise or counter-clockwise 3. Degree or radian 4. Click "OK"
25	Góc giữa ba điểm		Chọn ba điểm
	Angle formed from three points		Choose three points
26	Góc giữa hai đường thẳng		Chọn hai đường thẳng muốn tìm góc giữa
	Angle formed from two lines		Choose two lines
27	Đa giác		Chọn các điểm thuộc đa giác, kết thúc bằng cách chọn lại điểm đầu tiên
	Polygon		Choose all the points - vertices of the polygon, end at the first point
28	Đa giác đều		<p>Vẽ đa giác đều</p> <p>Nhập số cạnh đa giác</p> <p>3</p> <p>Chọn</p> <p>Đóng</p>
	Regular Polygon		Chọn hai điểm làm cạnh của đa giác rồi nhập số cạnh muốn tạo
29	Hình dịch		Chọn một hình muốn vẽ hình dịch rồi chọn véc tơ tính tiến
	Translated figure		Choose a figure and a vector
30			<p>Vẽ góc với điều chỉnh nhập liệu</p> <p>Nhập số đo góc</p> <p>1 90</p> <p>2</p> <p>3</p> <p>4 OK</p> <p>Độ</p> <p>Radian</p> <p><input type="checkbox"/> x n</p> <p>Hủy</p>

	Hình xoay		<p>Chọn hình muốn vẽ hình xoay, chọn một điểm làm tâm xoay rồi nhập góc muôn xoay</p> <ol style="list-style-type: none"> 1. Góc xoay 2. Chọn xoay theo chiều kim đồng hồ hoặc ngược chiều kim đồng hồ 3. Chọn đơn vị tính số đo góc là độ hay radian 4. Bấm "OK"
	Rotated figure		<p>Enter the rotated angle magnitude, choose a figure and a point as a center to rotate</p> <ol style="list-style-type: none"> 1. Angle magnitude 2. Clockwise or counter-clockwise 3. Degree or radian 4. Click "OK"
31	Hình đối xứng qua điểm		Chọn một hình và một điểm qua đó vẽ đối xứng
	Reflected by point figure		Choose a figure and a point through which draw a reflection
32	Hình đối xứng qua đường thẳng		Chọn một hình và một đường thẳng qua đó vẽ đối xứng
	Reflected by line figure		Choose a figure and a linear object through which draw a reflection
33	Vẽ liên tục		<p>Chức năng hỗ trợ người dùng vẽ một hình nhiều lần liên tiếp. Bấm vào để kích hoạt hoặc tắt. Màu đỏ tức là chưa kích hoạt, xanh dương là đang kích hoạt.</p>
	Continuous drawing		Click on the icon to continue drawing a figure many times. If the icon is blue, the feature is active. If it is red, it is inactive.

- **Cách dùng hộp công cụ - Using drawing pad**

- Vẽ hình mới – Draw new figure



- Chức năng vẽ hình hỗ trợ người dùng thực hiện việc vẽ một cách nhanh chóng và linh hoạt nhất:
 1. Chọn hình muốn vẽ
 2. Thanh thông báo sẽ hiện lên hướng dẫn
 3. Vẽ theo sự hướng dẫn

- This feature supports user to create a new figure easily and conveniently:
 1. Choose figure
 2. The notification bar will give user the info of how to draw
 3. Select of draw as guide

- **Di chuyển hình – Move figure**

- Để di chuyển một hình đã vẽ, ta bấm và giữ chuột trái trên hình đó rồi kéo hình đến vị trí mong muốn sau đó thả ra. Vị trí hình sau khi di chuyển sẽ được lưu vào lịch sử và cho phép người dùng quay lui về vị trí cũ nếu cần.

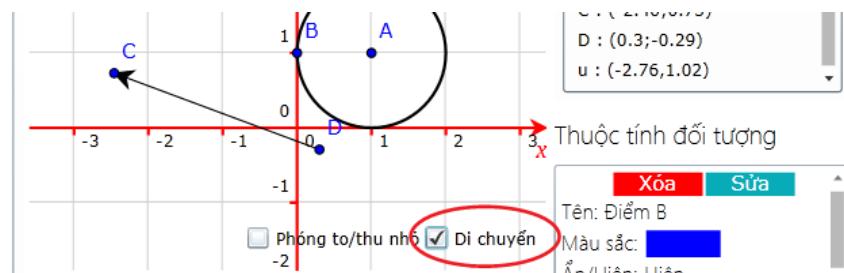
- To move a figure, user clicks and holds on the figure, then drag it to another location before drop it. The new location will be saved in history, which permit user to undo if needed.

- **Phóng to & thu nhỏ - Zoom in & zoom out**



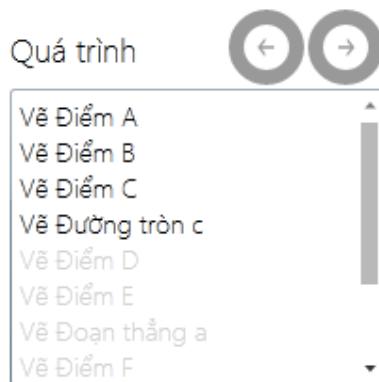
- Để phóng to và thu nhỏ tỉ lệ bản vẽ, ta đánh dấu vào ô "Zoom", sau đó cuộn chuột lên xuống.
- Người dùng cũng có thể dùng biểu tượng để phóng to và để thu nhỏ. Giá trị trên trục sẽ thay đổi tùy theo tỉ lệ ta phóng to hay thu nhỏ.
- To enable zoom feature with mouse, check "Zoom" checkbox, then scroll the mouse up to zoom in or down to zoom out.
- User can also click the icon to zoom in and icon to zoom out. The values on axis will also change according to the ratio.

○ Di chuyển màn hình vẽ - Move screen



- Để di chuyển màn hình vẽ, ta đánh dấu vào ô "Move", sau đó chọn một khoảng trống trên bảng vẽ và bấm giữ chuột trái rồi kéo.
- To move the screen, check the "Move" checkbox, and then click on a blank area in the drawing screen to drag and drop it.

• Quá trình/Lịch sử vẽ - History



- Trong từng trang, Pythagore cung cấp hộp thoại Quá Trình lưu lại toàn bộ lịch sử vẽ theo trình tự thời gian, đồng thời cung cấp khả năng Quay Lại hoặc Tiến Tới, giúp người dùng quản lý các hành động của mình thuận lợi, dễ

- In each sheet, Pythagore provides a history box to save all the actions performed in time order: draw, move, change properties... This helps user to

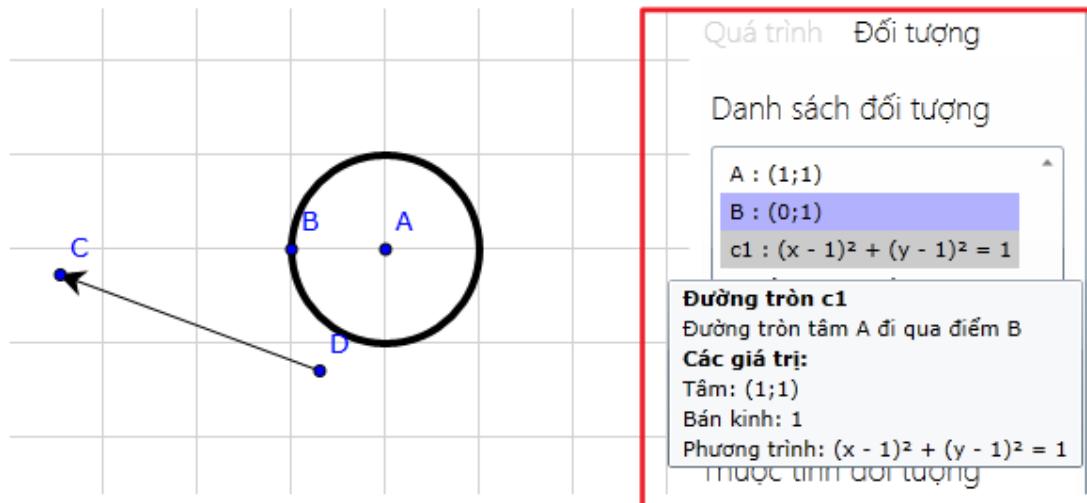
dàng. Người dùng có thể quay lui về hành động trước bằng cách chọn dấu mũi tên hoặc phím tắt Ctrl+Z, hoặc tiến tới hành động kế tiếp bằng hoặc phím tắt Ctrl+Y.

- Hành động lưu xuống lịch sử sẽ có màu đen và hành động đã được quay lui sẽ chuyển sang màu xám.

undo and redo their actions easily. User can undo by clicking on or hot key Ctrl+Z, redo by clicking or hot key Ctrl+Y.

- Actions that can be undone will be colored in black, and actions that can be redo will be colored in grey.

- **Danh sách đối tượng – Object list**



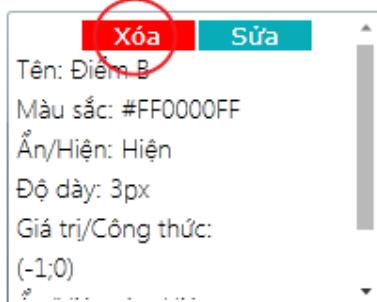
- Pythagore cũng cung cấp hộp thoại "Danh sách đối tượng", nhằm lưu lại các hình vẽ cũng như giá trị của chúng. Khi rê chuột lên trên một đối tượng, chương trình sẽ hỗ trợ tooltip miêu tả hình vẽ, đồng thời tập trung vào hình vẽ đó.

- Pythagore also provides a Property box to display all drawn figures and their values. When the mouse pointer hovers on an object, a tooltip will appear with additional information, at the same time, that figure also is highlighted.

- **Bảng thuộc tính đối tượng – Property panel**

- Xóa đối tượng – Delete object

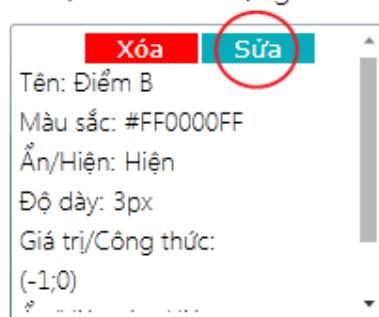
Thuộc tính đối tượng



- Để xóa một đối tượng, người dùng chọn đối tượng đó. Sau khi đối tượng đã được chọn, hộp thoại "Thuộc tính đối tượng" sẽ hiện lên thông tin của đối tượng đó. Ta bấm vào nút "Xóa" hoặc bấm Del.
- To delete an object, user chooses that object before going to the property box and click "Xóa" button, or simply click Del button.

- Chỉnh sửa thuộc tính đối tượng – Edit object's properties

Thuộc tính đối tượng



Thuộc tính đối tượng



- Để thay đổi thuộc tính một đối tượng, ta chọn đối tượng đó, sau đó vào bảng "Thuộc tính đối tượng" bấm vào nút "Sửa":

1. Đổi tên đối tượng. Lưu ý, tên đối tượng không được trùng với tên đã có sẵn.
2. Đổi màu sắc đối tượng. Sau khi bấm vào, chương trình sẽ cung cấp cho ta bảng màu để chọn màu cho đối tượng
3. Chọn thuộc tính Ẩn hay Hiện cho đối tượng. Nếu chọn ẩn, đối tượng sẽ trở nên vô hình trên bảng vẽ

- To change the property of an object, first choose that object, then go to the Property box, click on "Sửa" button:

1. Change object's name. The name cannot be existed on the drawing panel.
2. Change object's color. After clicking, a color table will appear for user to choose.
3. Hidden/Show object. If checked, object will be invisible on the screen.

4. Đổi độ dày cho viền đối tượng.
Lưu ý độ dày chỉ có thể trong khoảng 0-10
 5. Cho phép đổi tượng Ẩn hay Hiện tên trên bảng vẽ
 6. Chọn loại đường viền cho đối tượng (nét liền/gạch/chấm gạch)
 7. Chọn nút "Lưu".
- Đối tượng sẽ thay đổi thuộc tính theo những thay đổi người dùng đã thực hiện

4. Change object's thickness. The value can only be between 0-10.
 5. Hidden/Show object name on screen.
 6. Choose border style (solid, dashed, dot dashed)
 7. Choose "Lưu" button.
- Object will change according to the change made.

Thông báo - Notification

- Thanh thông báo nằm phía trên cùng của trang. Pythagore dùng thanh này để thông báo tất cả các thông tin trong phòng
 - Thông báo có ba loại cấp độ: Thông tin, Cảnh báo hoặc Lỗi:
- **Thông tin:** Thanh thông báo sẽ có màu xanh dương. Thông tin sẽ thông báo cho người dùng tình trạng của phòng, lưu và tải trang, thành viên mới tham gia, giơ tay, trao quyền... Thanh thông tin còn hướng dẫn người dùng trình tự vẽ hình để hạn chế tối đa việc xảy ra lỗi trong quá trình vẽ.



- **Cảnh báo:** Thanh thông báo màu vàng. Cảnh báo dùng để thông báo khi phòng đang trong quá trình lưu dữ liệu mới. Nếu cảnh báo hiện lên quá lâu thì có thể có lỗi xảy ra trong quá trình lưu, người dùng có thể làm mới trang để tiếp tục công việc.
- **Warning:** The notification bar's color is yellow. This info will appear when a room is in process of save new data. If this bar appears longer than 10 seconds, please refresh the page to continue the work.



➤ **Lỗi:** Thanh thông báo màu đỏ kèm theo thông tin về lỗi: không kết nối được tới trang chủ, không tải được phòng, lỗi tải trang lên... Khi xảy ra lỗi, các hành động có thể không cập nhật được về máy chủ. Người dùng nên làm mới trang để tiếp tục công việc.



➤ **Error:** The notification bar's color is red. The error may include: cannot connect to server, cannot load room, error when download... When error happens, the actions may not be committed to the server. Please refresh the page to continue the work.

Phần 4 – Những lỗi có thể xảy ra – Errors that may occur

- Bạn có khả năng không xem được trang web của chúng tôi nếu trình duyệt của bạn chưa được cài đặt Silverlight.
- Bạn có thể không nhận hết thông tin nếu mạng của bạn không ổn định.
- Bạn có thể không thấy hết nội dung trang web nếu màn hình của bạn có độ phân giải quá nhỏ.
- You might not see our website if your browser doesn't have Silverlight installed.
- You cannot receive all the information needed if your internet is not stable enough.
- You cannot see the entire page content if your screen resolution is too low.

Appendix

Supported Drawing tools

Table 9 – Supported Drawing tools

Shape	Type of definition	Input	Constraints
Point	Free point	coordinate of x and y	
	Point on object	An object and a point	
Mid-point	Mid-Point of a Segment	1 segment	
	Mid-Point of two given points	2 points	2 points are unique
Intersection	Intersection of two conics	2 conics	
	Intersection of a line and a conic	A line and a conic	
	Intersection of two lines	2 lines	2 lines are unique
Segment	Segment formed from two points	2 points	2 points are unique
Vector	Vector of two points	2 points	2 points are unique
Ray	Ray formed from two points	2 points	2 points are unique
Polygon	Normal polygon	A list of points	
	Regular polygon	2 points and the number of edges	
Line	Line formed by two points	2 points	2 points are unique
	Line parallel to a linear object and go through a point	A linear object and a point	The point must not be on the object.
	Tangent of conics	A point and a conic	
Perpendicular line	Perpendicular to line through a point	A line and a point	
Perpendicular Bisector	Perpendicular bisector of two	2 points	2 points are unique

	points		
	Perpendicular bisector of a segment	A segment	
Angle Bisector	Angle bisector of three points	3 points	3 points are unique
	Angle bisector of two lines	2 lines	2 lines are unique
Arc	Semicircle	2 points	2 points are unique
	Circular arc	3 points	3 points are unique
	Circumcircular arc	3 points	3 points are unique
Angle bisector	Angle bisector of two lines	2 lines	2 lines are unique
Parabola	Parabola with given focus and directrix	A point and a directrix	
Circle	Circle with three points	3 points	3 points are unique and must not be straight
	Circle with two points	2 points	2 points are unique
	Circle with radius	A point and a radius	
Ellipse	Ellipse with three points	3 points	3 points are unique
Hyperbola	Hyperbola with three points	3 points	3 points are unique
Angle	Angle with three points	3 points	3 points are unique
	Angle with two points	2 points	2 points are unique
	Angle with two lines	2 lines	2 lines are unique

Task sheets

Table 10 – Task sheets

Task	Tùng	Châu	Quang	Nhật
Technology feasibility	Runnable demo		GUI prototype, Icons	
Theoretical analysis		Math lessons		Math lessons
Module Figures	interface, base class, Segment, Group, Polygon, Angle	Conic	Arc, Ray, Vector	Line, Point
Module Actions	interface, base class, ActionManager Create, move, remove, property change		Rename Action	
Module Transformation	Transformation formula	Transformation class		
Module Visual	Base class, point, circle, segment, group, polygon	Ellipse	Ray, Angle, Vector	Line, Hyperbola, Parabola
Module Sequence	base classes, Point, Segment, Circle	Arc, Intersection, Ray, Vector, MidPoint, Transformation		Angle, Line, Conic
Database	All			
Client	DrawingControl, Toolbox	Login, Logout, Register, Upload, Download XML	HomeScreen, RoomScreen, ShareClient, Chat box	Edit information, Zoom, Move control, Capture Screen
Service	Notification service	Chat, Account Service	Share service	Room Service
RealTime	History, Notification	Serialize and Deserialize figures		
Report 1	All			
Report 2	problem definition, coding convention	Task, Task sheet, Meeting minute	Project organization	Review

Report 3	Entity Relation Diagrams, Review, Revise	Use case: Draw, Chat Features	User interface, Use case: Account, Chat	Use case: Account, Room, Sheet, Share
Report 4	Architecture overview, Component Description, Review, revise	Class diagrams, sequence diagram	Sequence diagrams	Component description
Report 5	Revise	Test cases		Test cases
Report 6	Deployment Guide	User Manual: Chat , Draw, Sheet, Notification features	Minimum system requirement, Overview, Account & Contact, Errors may occur part of User manual	User Manual: Room, Sheet, Share features

Tables

Table 1 - Roles and Responsibilities	3
Table 2 - Tasks.....	4
Table 3 - User's Possibilities.....	7
Table 4 - Screens.....	8
Table 5 - Actors.....	8
Table 6 – Other interfaces.....	47
Table 7 - Toolbar Icon Description.....	150
Table 8 - Drawing Toolbox Icon Description.....	153
Table 9 – Supported Drawing tools	A
Table 10 – Task sheets.....	C

Figures

Figure 1 – Overall Use-Case diagram	9
Figure 2 – Account feature use-case diagram	10
Figure 3 – Chat feature use-case diagram.....	13
Figure 4 – Room management feature use-case diagram.....	15
Figure 5 – Sheet Management Feature Use-case diagram	20
Figure 6 - Contact Management feature use-case diagram.....	24
Figure 7 - Sharing feature use-case diagram.....	26
Figure 8 - Drawing feature use-case diagram.....	31
Figure 9 - Pythagore Architecture	36
Figure 10 - Pythagore Component Diagram.....	38

Figure 11 - Overview class diagram for Common library.....	39
Figure 12 - Common.Actions class diagram.....	40
Figure 13 - Common.Figures namespace class diagram – part 1 of 2	43
Figure 14 - Common.Figures namespace class diagram – part 2 of 2	44
Figure 15 - Silverlight Library Component Overview.....	60
Figure 16 - SilverlightLibrary.Visual Namespace.....	61
Figure 17 - SilverlightLibrary.DrawingSequences class diagram.....	63
Figure 18 - Pythagore Services class diagram	71
Figure 19 - Silverlight client class diagram	76
Figure 20 - Drawing figures sequence diagram.....	77
Figure 21 - Undo sequence diagram	78
Figure 22 - Redo sequence diagram	78
Figure 23 - Save state sequence diagram	79
Figure 24 - Load State sequence diagram	80
Figure 25 - Export sheet sequence diagram.....	81
Figure 26 - Remove room sequence diagram	82
Figure 27 - Register screen	83
Figure 28 - Home screen: room tab	84
Figure 29 - Home screen: share tab	85
Figure 30 - Share object when clicked.....	85
Figure 31 - Add contact panel and group panel	86
Figure 32 - Edit information screen	86
Figure 33 - Room screen.....	87
Figure 34 - Pythagore Database Diagram.....	88
Figure 35 - PythagoreServer Application Pool	133
Figure 36 - PythagoreClient Application Pool	133
Figure 37 - PythagoreServer website configuration.....	134
Figure 38 - PythagoreClient website configuration.....	134
Figure 39 - IIS Configuration step complete	135